First Responders Guide to Computer Forensics

Richard Nolan
Colin O’Sullivan
Jake Branson
Cal Waits

March 2005

CERT Training and Education

HANDBOOK
CMU/SEI-2005-HB-001

Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

First Responders Guide to Computer Forensics

CMU/SEI-2005-HB-001

Richard Nolan
Colin O'Sullivan
Jake Branson
Cal Waits

March 2005

CERT Training and Education

This report was prepared for the

SEI Joint Program Office
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Christos Scondras
Chief of Programs, XPK

This work is sponsored by the SEI FFRDC primary sponsor and the Commander, United States Army Reserve (USAR) Information Operations Command and USAR EIO. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

List of Figures

List of Tables

Preface

This handbook targets a critical training gap in the fields of information security, computer forensics, and incident response. In today's networked world, it is essential for system and network administrators to understand the fundamental areas and the major issues in computer forensics. Knowledgeable first responders apply good forensic practices to routine administrative procedures and alert verification, and know how routine actions can adversely affect the forensic value of data. This awareness will greatly enhance system and network administrators' effectiveness when responding to security alerts and other routine matters. This capability is a crucial and an often overlooked element of defense-in-depth strategies for protecting the availability, integrity, and survivability of IT and network infrastructures. For instance, the step of collecting data from a live system is often skipped because of time constraints, lack of preparation, and the common practice of returning the corrupted live system to its original state by either a fresh software installation or a system reboot.

This handbook is designed to familiarize experienced system and network computer professionals with the fundamental elements of computer forensics and build on their existing technical skill set. The experienced security professional will encounter little in the way of new technical information contained in this material, but will discover new approaches and methods. The primary objective is to educate and motivate system and network administrators to approach both routine and unusual events in a safe forensic manner. When they understand the importance of their role as a first responder, the safety and effectiveness of an organization's use of computer forensics will be greatly enhanced. The authors would like to thank Bill Spernow of Experian for taking the time to review this handbook and share useful insights about the handbook's intended audience.

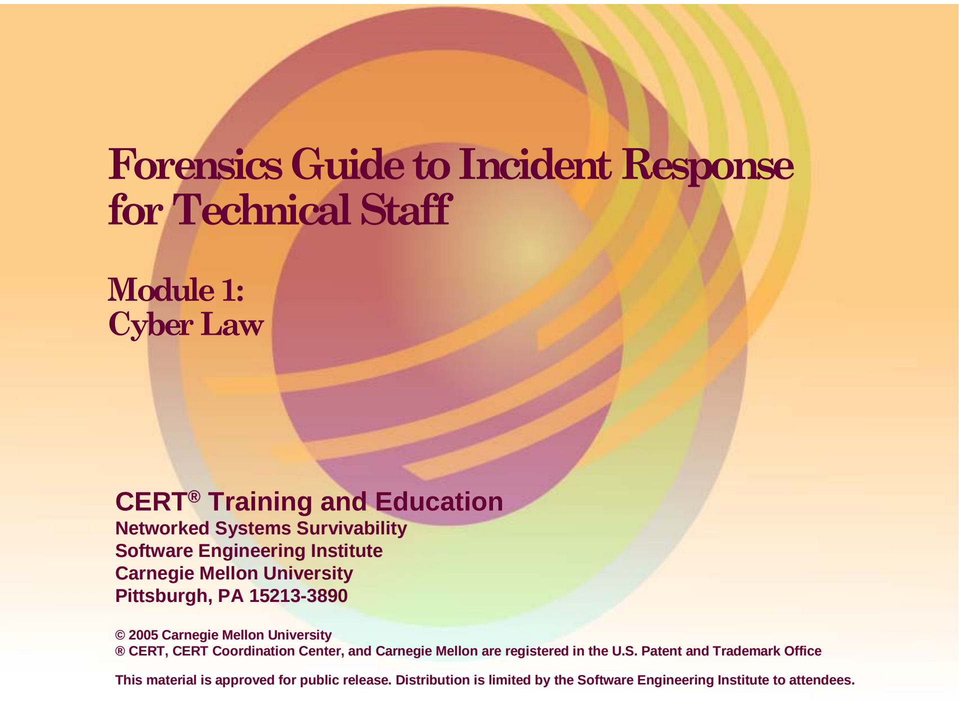
Abstract

This handbook is for technical staff members charged with administering and securing information systems and networks. It targets a critical training gap in the fields of information security, computer forensics, and incident response: performing basic forensic data collection. The first module describes cyber laws and their impact on incident response. The second module builds understanding of file systems and outlines a best practice methodology for creating a trusted first responder tool kit for investigating potential incidents. The third module reviews some best practices, techniques, and tools for collecting volatile data from live Windows and Linux systems. It also explains the importance of collecting volatile data before it is lost or changed. The fourth module reviews techniques for capturing persistent data in a forensically sound manner and describes the location of common persistent data types. Each module ends with a summary and a set of review questions to help clarify understanding.

This handbook was developed as part of a larger project. The incorporated slides are from the five day hands-on course *Forensics Guide to Incident Response for Technical Staff* developed at the SEI. The focus is on providing system and network administrators with methodologies, tools, and procedures for applying fundamental computer forensics when collecting data on both a *live* and a *powered off machine*. A live machine is a machine that is currently running and could be connected to the network. The target audience includes system and network administrators, law enforcement, and any information security practitioners who may find themselves in the role of first responder. The handbook should help the target audience to

- understand the essential laws that govern their actions
- understand key data types residing on live machines
- evaluate and create a trusted set of tools for the collection of data
- collect, preserve, and protect data from live and powered off machines
- learn methodologies for collecting information that are forensically sound (i.e., able to withstand the scrutiny of the courts)

1 Module 1: Cyber Law



Forensics Guide to Incident Response for Technical Staff

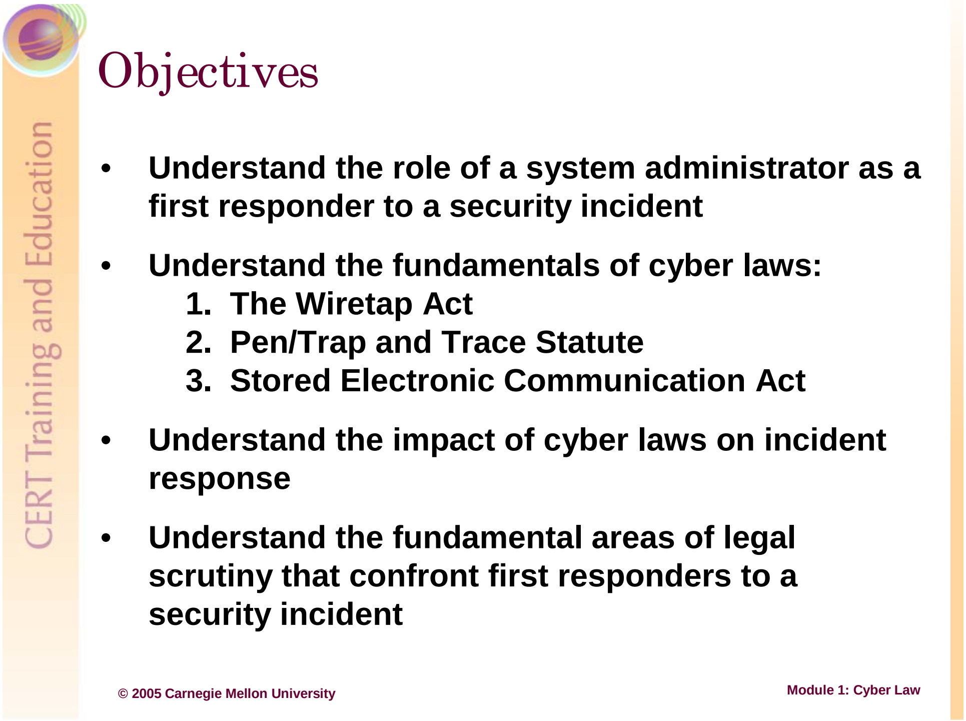
Module 1:
Cyber Law

CERT® Training and Education
Networked Systems Survivability
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

This material is approved for public release. Distribution is limited by the Software Engineering Institute to attendees.

Module 1: Cyber Law discusses aspects of federal laws that affect computer security professionals. The objective is to motivate security professionals to frame and consider their technical actions in the context of existing legal governance. Throughout this module several complex legal issues are reviewed briefly. While many of these issues could occupy an entire course, our goal is to expose the reader to topical areas of cyber law and raise awareness of how these laws could potentially impact their actions.

Additionally, many scenarios are presented in a technical setting that demonstrates the impact of cyber laws. Scenarios are meant to help the reader understand the potential application and implications of cyber laws, and not as a definitive interpretation of the law. Every effort has been made to represent the laws accurately; however, this document is not an authoritative legal guide.



CERT Training and Education

Objectives

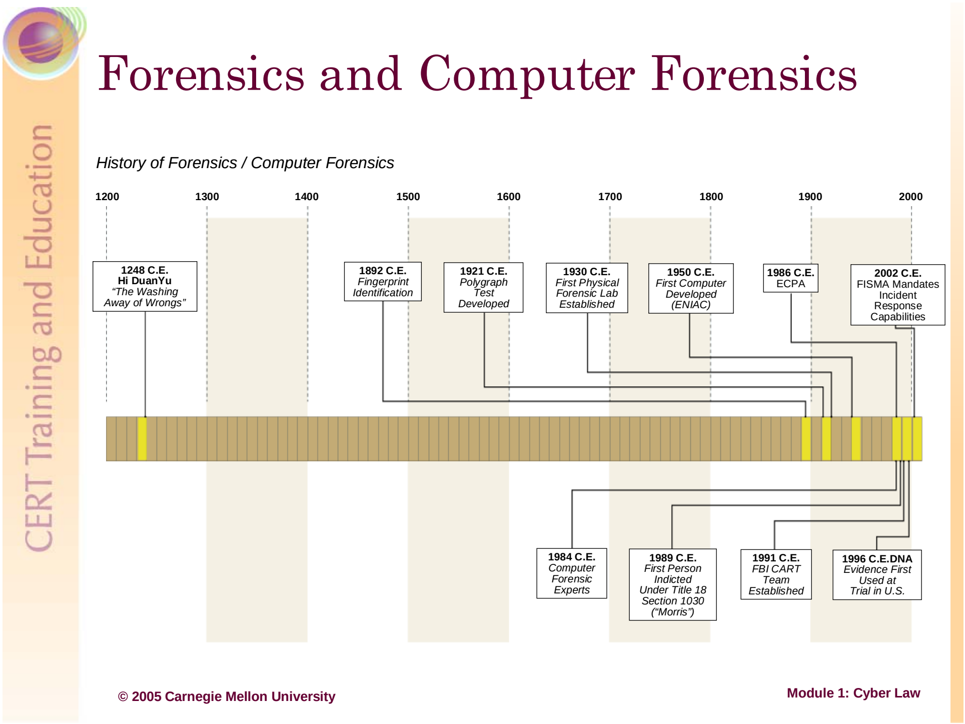
- Understand the role of a system administrator as a first responder to a security incident
- Understand the fundamentals of cyber laws:
    1. The Wiretap Act
    2. Pen/Trap and Trace Statute
    3. Stored Electronic Communication Act
- Understand the impact of cyber laws on incident response
- Understand the fundamental areas of legal scrutiny that confront first responders to a security incident

© 2005 Carnegie Mellon University

Module 1: Cyber Law

1.1 Module 1 Objectives

The above slide lists the four primary objectives for Module 1, “Cyber Law.”



CERT Training and Education

Forensics and Computer Forensics

History of Forensics / Computer Forensics

1200 1300 1400 1500 1600 1700 1800 1900 2000

1248 C.E.
Hi DuanYu
"The Washing Away of Wrongs"

1892 C.E.
Fingerprint Identification

1921 C.E.
Polygraph Test Developed

1930 C.E.
First Physical Forensic Lab Established

1950 C.E.
First Computer Developed (ENIAC)

1986 C.E.
ECPA

2002 C.E.
FISMA Mandates Incident Response Capabilities

1984 C.E.
Computer Forensic Experts

1989 C.E.
First Person Indicted Under Title 18 Section 1030 ("Morris")

1991 C.E.
FBI CART Team Established

1996 C.E.DNA
Evidence First Used at Trial in U.S.

© 2005 Carnegie Mellon University

Module 1: Cyber Law

1.2 Forensics

Forensics is the process of using scientific knowledge in the collection, analysis, and presentation of evidence to the courts. The word *forensics* means "to bring to the court." Forensics deals primarily with the recovery and analysis of latent evidence. Latent evidence can take many forms, from fingerprints left on a window to DNA evidence recovered from blood stains. Forensics takes scientific knowledge and methods and combines them with standards of law for the purpose of recovering and presenting at court evidence related to a crime.

The field of forensics has developed over many centuries. The earliest record of forensics dates back to 1248 C.E., when a Chinese physician named Hi Duan Yu wrote *The Washing Away of Wrongs*. Yu presented the anatomical and medical knowledge of the time as it related to law, such as the difference between drowning, strangulation, and death by natural causes.[1] Fingerprint identification was not used and understood until 1892. The first forensic lab was established in the United States in 1930. DNA evidence was first used as evidence at a trial in 1996.[2]

Although forensics has developed into a well-documented and disciplined field of study with many levels of certifications, it is always changing. As scientific knowledge and practices expand, so too will the processes of forensics. Nevertheless, forensics will always lag behind

[1] http://www.mdpd.com/astmhtm.html

[2] http://www.crimtrac.gov.au/dnahistory.htm

the advances of science in some way. An example is DNA evidence—DNA typing was understood and utilized in the 1980s in New Zealand, but did not appear as evidence in the U.S. courts until 1996.

Considering that modern forensics has had almost six decades to develop in the U.S. courts and 800 years as a discipline, the specialized field of computer forensics, in comparison, is still in its infancy. It should come as no surprise, then, that there is very little standardization and consistency across the courts and industry in the area of computer forensics, as a result it is not yet recognized as a formal "scientific" discipline.

1.2.1 Computer Forensics

Computer forensics can be defined as the collection and analysis of data from computer systems, networks, communication streams (wireless) and storage media in a manner that is admissible in a court of law. It is a merger of the disciplines of computer science and the law. This definition applies to the collection of information in real time, as well as the examination of latent data.

Here is what others have to say about the definition of computer forensics:

> *"Computer forensics" can thus not afford solely to concern itself with procedures and methods of handling computers, the hardware from which they are made up and the files they contain. The ultimate aim of forensic investigation is use in legal proceedings [Mandia 01].*
>
> *The objective in computer forensics is quite straightforward. It is to recover, analyze and present computer based material in such a way that it is useable as evidence in a court of law [Mandia 01].*
>
> *The key phrase here is "useable as evidence in a court of law." It is essential that none of the equipment or procedures used during the examination of the computer obviate this single requirement.*[3]

At one time, computer forensics dealt exclusively with *persistent data*—data that is stored on a local hard drive or other medium and is preserved when the computer is powered off—and was the sole domain of law enforcement. In this age of interconnected networks, however, the examination and collection of *volatile data* is becoming more important. Volatile data is any data that is stored in memory, or in transit, that will be lost when the computer loses power or is powered off. Volatile data resides in registries, cache, and random access memory (RAM). Since the nature of volatile data is effervescent, collection of this information will likely need to occur in real or near-real time.

[3] http://www.dibsusa.com/methodology/methodology.html

Also, with the rapid expansion of the Internet and the introduction of recent legislation such as HIPAA,[4] Sarbanes-Oxley,[5] California Act 1798,[6] and others, all of which hold businesses civilly and, in some cases, criminally liable for breaches in the security or integrity of computer networks and systems, companies are not only trying to protect themselves from Internet-related threats but are also dealing with compliance and liability concerns related to these laws. Several current writings point to the fact that companies are allocating more of their IT budgets towards security.[7] In fact, International Data Corporation (IDC) has reported that the market size for intrusion-detection and vulnerability-assessment software will reach 1.45 billion dollars in 2006. These are relatively well-known trends in the IT community. These factors and others have increased the deployment of network security devices such as intrusion detection systems (IDS), firewalls, proxies, etc., which all report on the security status of networks to some degree in real time or near-real time. So much of security is focused on locking down and monitoring systems that an often overlooked area is how information and evidence related to the alerts are technically handled.

Almost always, when a security breach or monitoring alert occurs, some sort of verification needs to happen. No matter what type of alert is generated, verification is required to determine whether the alert is a false positive or is in fact an indication of some breach, attempted breach, or incident. This verification, often a form of routine administrative task, is almost always conducted on a live machine, and deals with volatile data. System administrators and security personnel must possess a basic understanding of how routine administrative tasks can affect both the forensic process (the potential admissibility at court) and the subsequent ability to recover data that can potentially be critical to the identification and analysis of a security incident.

Contrary to the impression given by popular television shows, there are not nearly enough forensic investigators available to respond to every crime scene to collect, process, and analyze evidence. As a result, many law enforcement officers are being trained to process and collect evidence (fingerprints, fibers, etc.) from crime scenes in a forensically sound manner. A qualified investigator then analyzes the evidence and makes conclusions at a later time. At trial, or before the courts, the collector authenticates the collected evidence and the expert testifies to its significance. This division of labor is also needed in the discipline of computer forensics. Like law enforcement officers, system administrators are often the “first responders” at potential “crime scenes.” It is critical that they possess the technical skills and ability to preserve critical information related to a suspected security incident in a

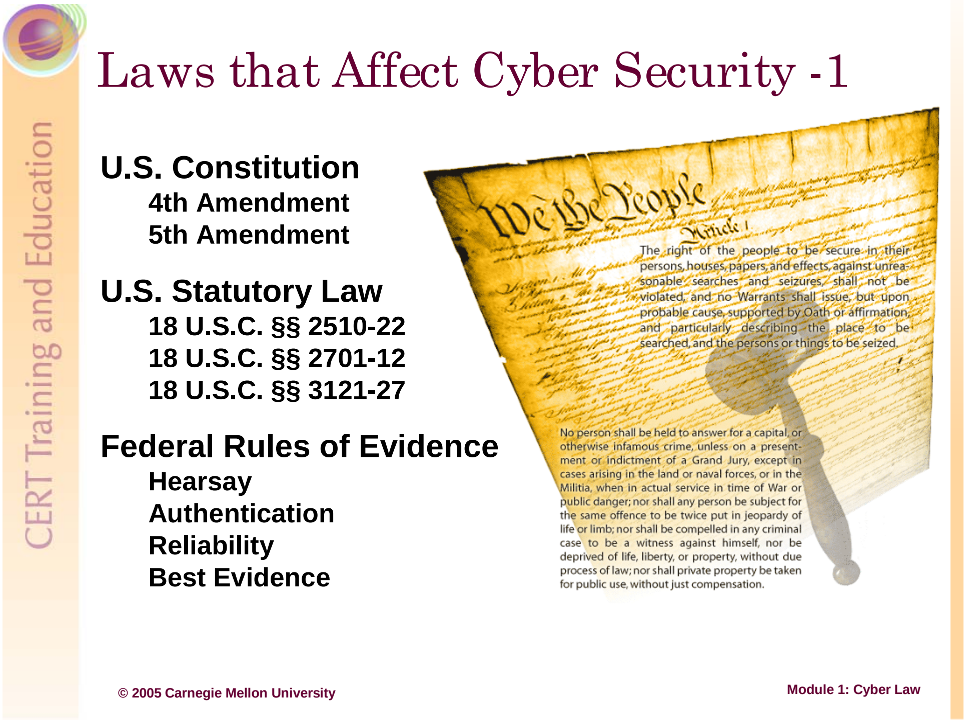
[4] Health Insurance Portability and Accountability Act of 1996 (see http://www.hhs.gov/ocr/hipaa/)

[5] Sarbanes-Oxley Act of 2002 (http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ204.107.pdf).

[6] Information Practices Act of 1977 (http://www.privacy.ca.gov/code/ipa.htm).

[7] http://www.vnunet.com/news/1150059,
http://itmanagement.earthweb.com/it_res/print.php/3085471,
http://www.csoonline.com/csoresearch/report50.html

forensically sound manner and that they be aware of the legal issues that govern their actions. This skill set is not easily acquired, and is the subject of many courses of study.


CERT Training and Education
We the People
Article 1

Laws that Affect Cyber Security -1

U.S. Constitution
- 4th Amendment
- 5th Amendment

U.S. Statutory Law
- 18 U.S.C. §§ 2510-22
- 18 U.S.C. §§ 2701-12
- 18 U.S.C. §§ 3121-27

Federal Rules of Evidence
- Hearsay
- Authentication
- Reliability
- Best Evidence

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

No person shall be held to answer for a capital, or otherwise infamous crime, unless on a presentment or indictment of a Grand Jury, except in cases arising in the land or naval forces, or in the Militia, when in actual service in time of War or public danger; nor shall any person be subject for the same offence to be twice put in jeopardy of life or limb; nor shall be compelled in any criminal case to be a witness against himself, nor be deprived of life, liberty, or property, without due process of law; nor shall private property be taken for public use, without just compensation.

Module 1: Cyber Law

1.3 Laws that Affect Cyber Security

Computer forensics is a relatively new discipline to the courts and many of the existing laws used to prosecute computer-related crimes, legal precedents, and practices related to computer forensics are in a state of flux. Almost daily, new court rulings are handed down that affect how computer forensics is applied. The best source of information in this area is the United States Department of Justice's Cyber Crime Web Site.[8] Here all the recent court cases can be reviewed, as well as numerous guides on the introduction of computer evidence in court and what standards apply.

One recent court case, United States Court of Appeals for the First Circuit, No. 03-1383 UNITED STATES OF AMERICA, Appellant, v. BRADFORD C. COUNCILMAN, Defendant, Appellee, has a couple of interesting aspects: (1) it is a landmark case in that it could fundamentally change the way emails are dealt with and (2) the courts acknowledge that they are "wrestling with the antiquated nature of some of the laws." Circuit Judge Juan R. Torruella wrote this in his summary:

> *Moreover, at this juncture, much of the protection [of the Wiretap Act] may have been eviscerated by the realities of modern technology. We observe, as most courts have, that the language may be out of step with the technological realities*

[8] http://www.cybercrime.gov

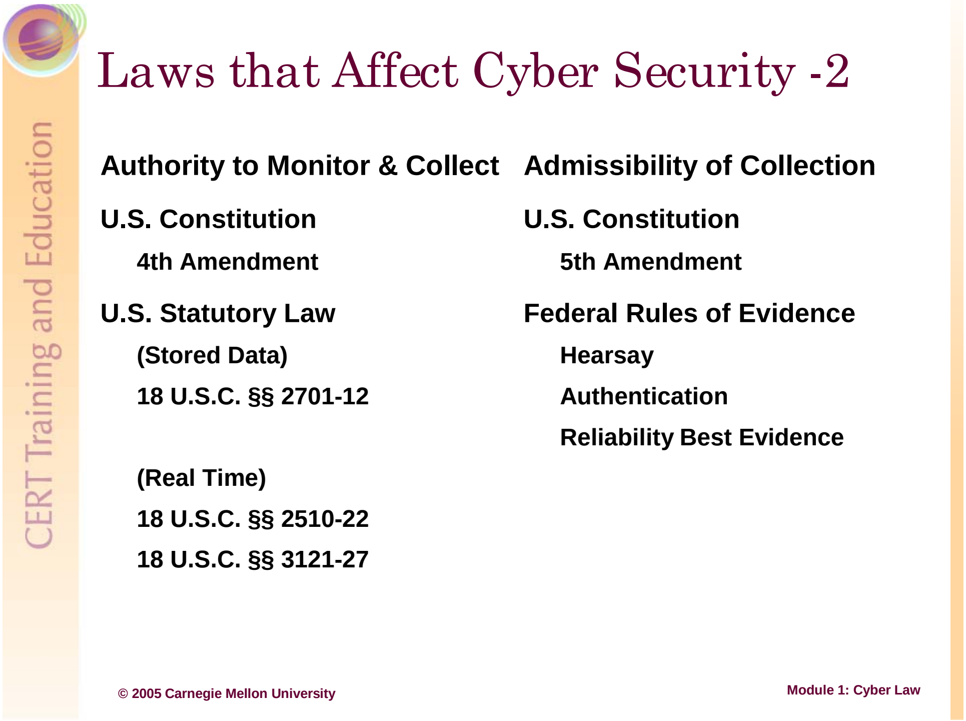
> *of computer crimes. This particular appeal reflects how the Courts are wrestling with the antiquated nature of some of the laws governing electronic communications.*[9]

The appeal was over whether email should be considered stored electronic communication or whether it should considered transmitted electronic communication. The outcome of this case was that the court did not believe that emails are protected by the Wiretap Act under the Store Communication Section when they are in temporary file storage at various locations along their delivery route. Although the court ruled in favor of the defendant, an interesting dissenting opinion was presented. Essentially, the dissenting judge uses sections of an RFC on emails[10] and other very technical information and expresses two things: (1) that the current law was never intended to be applied to this type of technology and (2) that the “intent” of the law only protects email while it is being transmitted from sender to receiver.

Again, keep in mind that this module is meant not to be a course in legal studies but rather to serve as an introductory exposure to major legal governance issues that affect system administrators and network security personnel. To use a training analogy from law enforcement, police officers are not lawyers, nor will their training ever make them so; however, police officers receive training in legal areas that authorize their actions and govern their responses. For example, police officers all have a fundamental understanding of the restrictions of the 4th and 5th Amendments of the U.S. Constitution and the several exceptions to them. This is because these two amendments have significant impact on how law enforcement officers conduct themselves. This same model needs to be applied to system administrators and network security personnel. This is not to say that they should try to function as law enforcement, but rather that they should be aware of and understand the fundamental legal issues governing their actions.

[9] http://www.ca1.uscourts.gov/cgi-bin/getopn.pl?OPINION=03-1383.01A

[10] RFC 821, by Jonathan B. Postel (see http://www.faqs.org/rfcs/rfc821.html).



CERT Training and Education

Laws that Affect Cyber Security -2

Authority to Monitor & Collect

U.S. Constitution

4th Amendment

U.S. Statutory Law

(Stored Data)

18 U.S.C. §§ 2701-12

(Real Time)

18 U.S.C. §§ 2510-22

18 U.S.C. §§ 3121-27

Admissibility of Collection

U.S. Constitution

5th Amendment

Federal Rules of Evidence

Hearsay

Authentication

Reliability Best Evidence

© 2005 Carnegie Mellon University

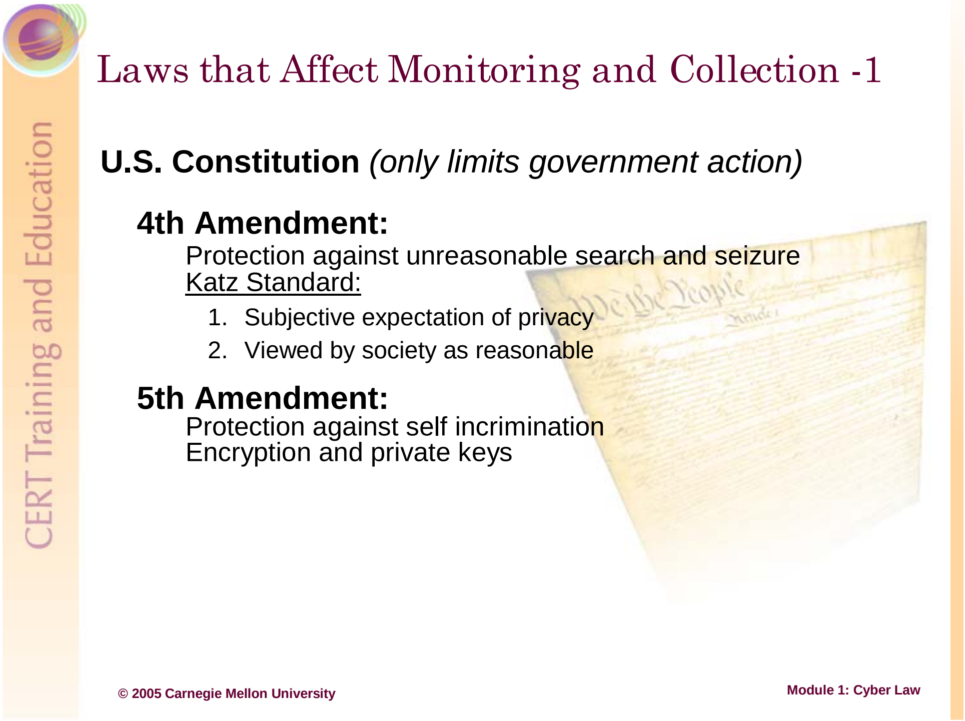
Module 1: Cyber Law

Three areas of law related to computer security will be examined:

1. U.S. Constitution
   - 4th Amendment “Protection against unreasonable search and seizure”
   - 5th Amendment “Protection against self incrimination”
2. U.S. Statutory Law
   - 18 U.S.C. 2510-22 (The Wiretap Act)
   - 18 U.S.C. 3121-27 (The Pen Registers and Trap and Trace Devices Statute)
   - 18 U.S.C. 2701-12 (The Stored Wired and Electronic Communication Act)
3. Federal Rules of Evidence
   - Hearsay
   - Authentication
   - Reliability
   - Best Evidence

This is not a comprehensive review of these legal areas but rather a summary review of key aspects that will likely affect the actions of system and network administrators.

In the United States of America, there are two primary areas of legal governance affecting cyber security actions related to the collection of network data: (1) authority to monitor and collect the data and (2) the admissibility of the collection methods. Of the three areas above, the U.S. Constitutional and U.S. Statutory Laws primarily govern the collection process, while the Federal Rules of Evidence deal primarily with admissibility.



Laws that Affect Monitoring and Collection -1

U.S. Constitution *(only limits government action)*

4th Amendment:

Protection against unreasonable search and seizure

Katz Standard:

1. Subjective expectation of privacy
2. Viewed by society as reasonable

5th Amendment:

Protection against self incrimination

Encryption and private keys

© 2005 Carnegie Mellon University

Module 1: Cyber Law

1.4 Legal Governance Related to Monitoring and Collection

A key point to remember when considering these amendments is that they were written some two hundred years ago and are now being applied to a technology that could not be imagined by the original authors.

1.4.1 Constitutional Issues

1.4.1.1 The 4th Amendment

The 4th Amendment to the United States Constitution is as follows:

> *The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.*

The 4th Amendment protects individuals from unreasonable searches and seizures conducted by agents of the government. *It does not extend protection from searches conducted by private parties who are not acting as agents of the government*. The question is whether or not your organization can be considered as “government”, or under the direction of a government agent, under the 4th Amendment. There isn’t always a *bright line* answer to that question (in law, a bright line is a rule that clearly and unambiguously delineates what is so and what is not). If in doubt, organizations should always seek the advice of legal counsel in determining the answer.

In general, the principles set forth by the 4th Amendment provide for individuals to enjoy a “reasonable expectation of privacy.” In determining what is a reasonable expectation of privacy, the U.S. courts rely in part on the Katz test (Katz v. United States, 389 U.S. 347, 362 (1967). The Katz test outlines two questions: (1) Does the individual’s conduct demonstrate a (subjective) expectation of privacy? and (2) Would society be prepared to recognize this as reasonable? These questions allow for a range of interpretations and are the subject of a significant amount of case law. However, they are the fundamental questions asked by the 4th Amendment. To explore a more detailed analysis of the issues surrounding the 4th Amendment, the following site is a good place to start: http://caselaw.lp.findlaw.com/data/constitution/amendment04/.

What system administrators and network security personnel need to understand is that individuals may enjoy a certain amount of privacy in the electronic world (emails, stored files, etc.). Some other key points are (1) the 4th Amendment does not apply to or restrict the conduct of private individuals, only that of the government; (2) the level of expected privacy is substantially less outside of the home, although not eliminated; and (3) protections (privacy) enjoyed from the 4th Amendment are dissolved when the item protected by this amendment is lost or otherwise ends up no longer in the custody of the owner. Also, if information or evidence is collected in a way that is determined to have violated the 4th Amendment, that information/evidence and all information/evidence subsequently obtained or discovered as a result of the original information/evidence will be suppressed, i.e., deemed inadmissible by the courts. This is known as the “fruit of the poisonous tree doctrine.”

The 4th Amendment has been applied to information technology in a number of cases. The following is an excerpt from a U.S. Department of Justice Manual:[11]

> *When confronted with this issue, courts have analogized electronic storage devices to closed containers, and have reasoned that accessing the information stored within an electronic storage device is akin to opening a closed container. Because individuals generally retain a reasonable expectation of privacy in the contents of closed containers, see United States v. Ross, 456 U.S. 798, 822-23 (1982), they also generally retain a reasonable expectation of privacy in data held within electronic storage devices. Accordingly, accessing information stored in a computer ordinarily will implicate the owner’s reasonable expectation of privacy in the information. See United States v. Barth, 26 F. Supp. 2d 929, 936-37 (W.D. Tex. 1998) (finding reasonable expectation of privacy in files stored on hard drive of personal computer); United States v. Reyes, 922 F. Supp. 818, 832-33 (S.D.N.Y. 1996) (finding reasonable expectation of privacy in data stored in a pager); United States v. Lynch, 908 F. Supp. 284, 287 (D.V.I. 1995) (same);*

[11] http://www.cybercrime.gov/s&smanual2002.htm

> *United States v. Chan, 830 F. Supp. 531, 535 (N.D. Cal. 1993) (same); United States v. Blas, 1990 WL 265179, at *21 (E.D. Wis. Dec. 4, 1990) ("[A]n individual has the same expectation of privacy in a pager, computer, or other electronic data storage and retrieval device as in a closed container.").*

1.4.1.2 The 5th Amendment

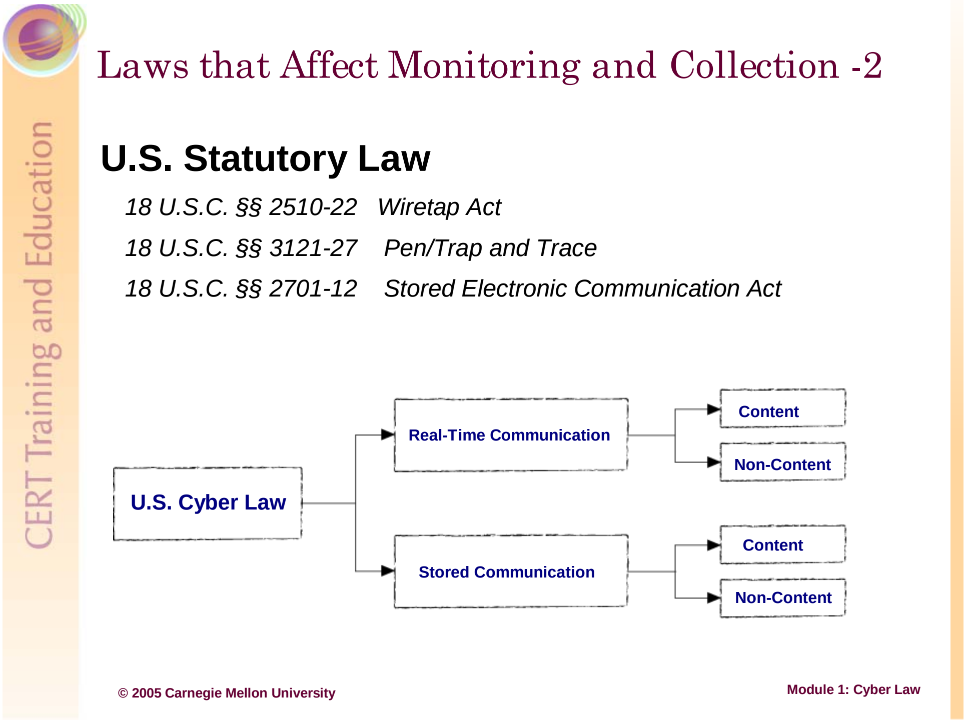
The 5th Amendment provides that

> *No person shall be compelled in any criminal case to be a witness against himself.*

The core issue related to computer forensics is that under current jurisprudence the 5th Amendment's protections against self incrimination extend to cryptographic keys. "Under the Fifth Amendment, an individual cannot be compelled to testify to his or her memorized key" [CDT 04]. The word "memorized" is an important qualifier, however, meaning that the key (passphrase) was never written down. Remember, the 5th Amendment only protects an individual from being compelled to provide incriminating *testimony*. Its protections do not extend to existing written and/or documentary evidence. This affects system or network administrators because if encrypted files are found during a collection or examination of a system or network, little can be done to force an individual to decrypt them.

For more details (in significant depth), visit the following Web sites:

- http://www.cdt.org/crypto/legis_105/mccain_kerrey/const_impact.html
- http://law.richmond.edu/jolt/v2i1/sergienko.html#h1



CERT Training and Education

Laws that Affect Monitoring and Collection -2

U.S. Statutory Law

18 U.S.C. §§ 2510-22 Wiretap Act

18 U.S.C. §§ 3121-27 Pen/Trap and Trace

18 U.S.C. §§ 2701-12 Stored Electronic Communication Act

U.S. Cyber Law

Real-Time Communication

Content

Non-Content

Stored Communication

Content

Non-Content

© 2005 Carnegie Mellon University

Module 1: Cyber Law

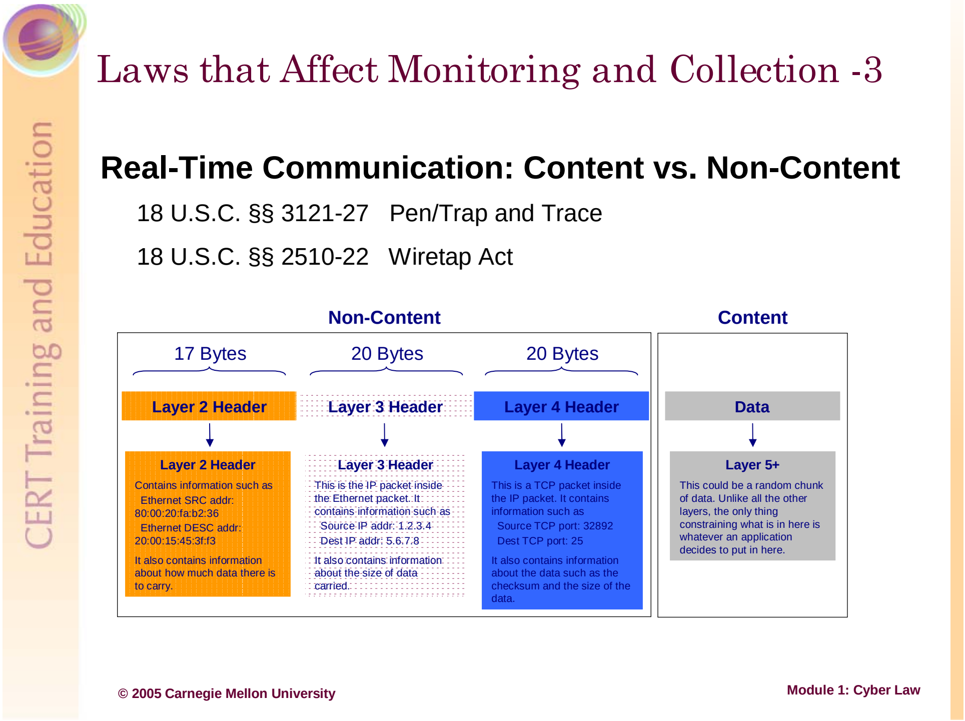
1.4.2 U.S. Statutory Law

There are three U.S. statutes that potentially impact the techniques used by system administrators and network security personnel during the collection of computer records and other digital evidence in response to a security incident:

• 18 U.S.C. §§ 2510-22	*Wire and Electronic Communications Interception and Interception of Oral Communications*
• 18 U.S.C. §§ 3121-27	*Pen Registers and Trap and Trace Devices*
• 18 U.S.C. §§ 2701-12	*Stored Wire and Electronic Communications and Transactional Records Access*

This section will highlight the intent and purpose of these statutes, as well as provide technical scenarios illustrating the exceptions to these statutes as they affect the actions taken by system administrators and network security personnel. Absent these exceptions, violations of these statutes could constitute a federal felony punishable by a fine and/or prison.

Of these three statutes, two deal with real-time electronic communications (18 U.S.C. §§ 2510-22 and 18 U.S.C. §§ 3121-27) and the other (18 U.S.C. §§ 2701-12) deals with stored electronic communications.



CERT Training and Education

Laws that Affect Monitoring and Collection -3

Real-Time Communication: Content vs. Non-Content

18 U.S.C. §§ 3121-27   Pen/Trap and Trace

18 U.S.C. §§ 2510-22   Wiretap Act

Non-Content

Content

17 Bytes

20 Bytes

20 Bytes

Layer 2 Header

Layer 3 Header

Layer 4 Header

Data

Layer 2 Header

Contains information such as
Ethernet SRC addr: 80:00:20:fa:b2:36
Ethernet DESC addr: 20:00:15:45:3f:f3

It also contains information about how much data there is to carry.

Layer 3 Header

This is the IP packet inside the Ethernet packet. It contains information such as
Source IP addr: 1.2.3.4
Dest IP addr: 5.6.7.8

It also contains information about the size of data carried.

Layer 4 Header

This is a TCP packet inside the IP packet. It contains information such as
Source TCP port: 32892
Dest TCP port: 25

It also contains information about the data such as the checksum and the size of the data.

Layer 5+

This could be a random chunk of data. Unlike all the other layers, the only thing constraining what is in here is whatever an application decides to put in here.

Module 1: Cyber Law

The Wiretap Act and Pen/Trap and Trace both deal with real-time communication, but they focus on very different aspects of that communication. It is important to note that both of these statutes are not clearly written to apply to the flow of data over the Internet, as they were born in an era when telecommunications was the primary medium of communication. They have been recently modified to better clarify their governance related to electronic communications, but there is little technical language embedded in the actual statutes themselves.

Below is a technical review of the Open System Interconnection (OSI) Model. Within the existing language of the above-mentioned statues the intent is that the Pen/Trap and Trace deals with Layer 2, 3, and 4 information and the Wiretap Act deals with Layer 5, 6, and 7 information.

Layer 1: **Physical**

This is the physical wire that carries the electronic pulses.

Layer 2: **Data Link** (17 bytes)

Contains information such as the media access control (MAC) addresses of the source sending the packet and the next hop or destination.

Layer 3: **Network** (20 bytes)

Contains information such as the source and destination IP addresses.

Layer 4: **Transport** (20 bytes)

Contains information such as source and destination ports.

Layer 5+: **Session/Presentation/Application**

This contains the data that the application requires (for example, this would be the data portion of an email message).

Often the Department of Defense (DoD) Model or Internet Protocol Suite and the Open System Interconnection (OSI) Model are used to describe the same process. Below is a mapping of the two models.

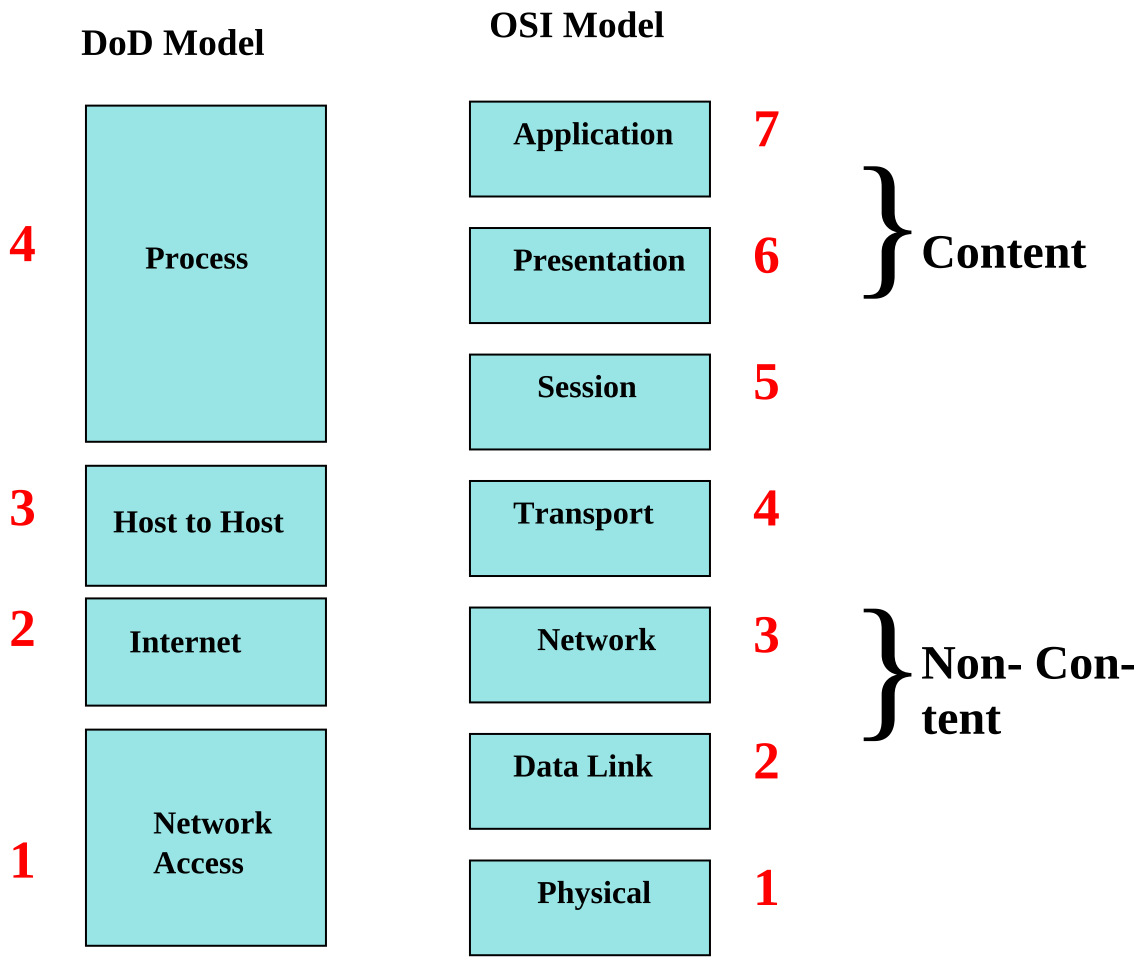
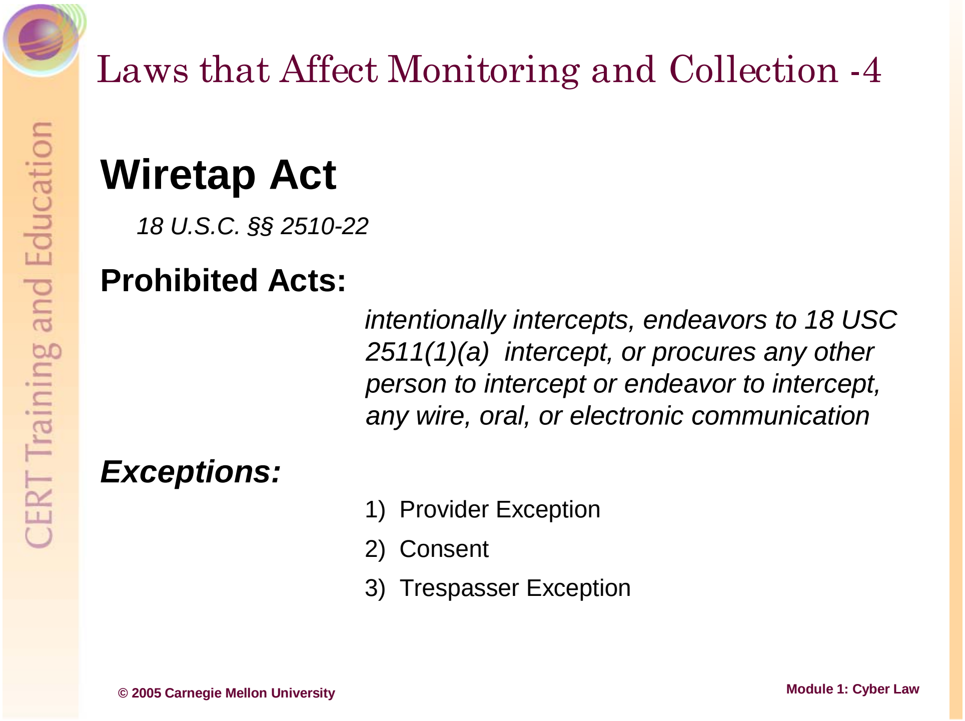


Figure 1: Mapping of DoD and OSI Models

It is important to understand the fundamental difference between what the laws consider “content” and “non-content.” Although there is no bright line, from a technical perspective the division occurs between Layer 4 (Layers 4 and below deal with routing, addressing, and signaling information) and Layer 5 (Layers 5 and above deal with the Data or Content).



Laws that Affect Monitoring and Collection -4

Wiretap Act

18 U.S.C. §§ 2510-22

Prohibited Acts:

intentionally intercepts, endeavors to 18 USC 2511(1)(a) intercept, or procures any other person to intercept or endeavor to intercept, any wire, oral, or electronic communication

Exceptions:

1) Provider Exception
2) Consent
3) Trespasser Exception

Module 1: Cyber Law

1.4.2.1 Wiretap Act/Electronic Communications Privacy Act

Generally speaking, the Wiretap Act, 18 U.S.C. §§ 2510-22, prohibits the interception of real-time electronic data communications (specifically the content), unless one of the statutory exceptions applies. There are several exceptions, but only two exceptions that potentially affect incident response activities will be considered in this documentation.

This law governs the usage of tools like sniffers (TCPDump, Ethereal, Etherpeek, etc.), which are used to intercept and examine the content (i.e., Layer 5+) of electronic data communication and any other device that looks at the content (Layer 7 firewalls, URL scanners, etc.). This law does not restrict the interception and examination of *signaling information* (information from the network and transport layers). Those actions are restricted by the Pen/Trap and Trace Statute, which is discussed later.

The Wiretap Act applies to anyone who

> *intentionally intercepts, endeavors to intercept, or procures any other person to intercept or endeavor to intercept, any wire, oral, or electronic communication.*

There are several exceptions enumerated in the statute. Of these, three have the greatest impact on the actions of system administrators and network security personnel.

1) Provider Exception, 18 U.S.C. § 2511(2)(a)(i)

> *It shall not be unlawful under this chapter for an operator of a switchboard, or an officer, employee, or agent of a provider of wire or electronic communication service, whose facilities are used in the transmission of a wire or electronic*

> *communication, to intercept, disclose, or use that communication in the normal course of his employment while engaged in any activity which is a necessary incident to the rendition of his service or to the protection of the rights or property of the provider of that service, except that a provider of wire communication service to the public shall not utilize service observing or random monitoring except for mechanical or service quality control checks.*

This allows for system administrators and network security personnel to conduct limited monitoring for the purpose of protecting the “rights or property” of the provider. It is permissible to monitor the content of electronic communications under this exception when there is a clear nexus between the threat and what is monitored.

Here is a scenario that demonstrates an example of a clear nexus between a threat to the “rights or property” of a provider and what is being monitored:

> In support of Web services that are critical to his operations, a system administrator is using a URL scanner. He is examining the “contents” of the Web requests, not just the connection-related information (Layers 3 and 4). A clear nexus exists between this monitoring of “content” and the protection of the Web server.[*]

2) Consent, 18 U.S.C. § 2511(2)(c)

> *…where such person is a party to the communication or one of the parties to the communication has given prior consent to such interception…*

Here is where bannering the network and written and signed acceptable use policies are critical. If you have a correctly worded banner and an acceptable use policy that informs users of monitoring activities and how the information gathered from monitoring will be used, in a vast majority of cases the consent burden will be satisfied.

An example of this would be when a company institutes a policy that forbids non-business-related Web surfing and states that monitoring of Web surfing will be conducted to ensure compliance. Here clear consent is required by the users of the system. Given this consent to a written or bannered policy, the company can intercept and monitor Web surfing activities to ensure compliance with the policy.[*]

Consent should never be taken as a general permission for system administrators to conduct ad hoc, random monitoring activities. Rather, monitoring activities related to consent should be part of a well-documented process that is explicitly enumerated in the obtained consent.

[*] This example is not intended to be legally authoritative but to frame a legal concept in terms that a system administrator will understand.

3) Trespasser Exception, 18 U.S.C. § 2510(21)

> *…a person who accesses a protected computer without authorization and thus has no reasonable expectation of privacy in any communication transmitted to, through, or from the protected computer…*

This is a very broad exception because of the definition of a "protected computer." Basically almost every public provider of electronic communication falls under this definition. Consultation with counsel should be sought to make an authoritative interpretation. The definition of "protected computer" is found in 18 U.S.C. 1030(21):[12]

> *The term "protected computer" means a computer*
>
> *(A) exclusively for the use of a financial institution or the United States Government, or, in the case of a computer not exclusively for such use, used by or for a financial institution or the United States Government and the conduct constituting the offense affects that use by or for the financial institution or the Government; or*
>
> *(B) which is used in interstate or foreign commerce or communication, including a computer located outside the United States that is used in a manner that affects interstate or foreign commerce or communication of the United States…*

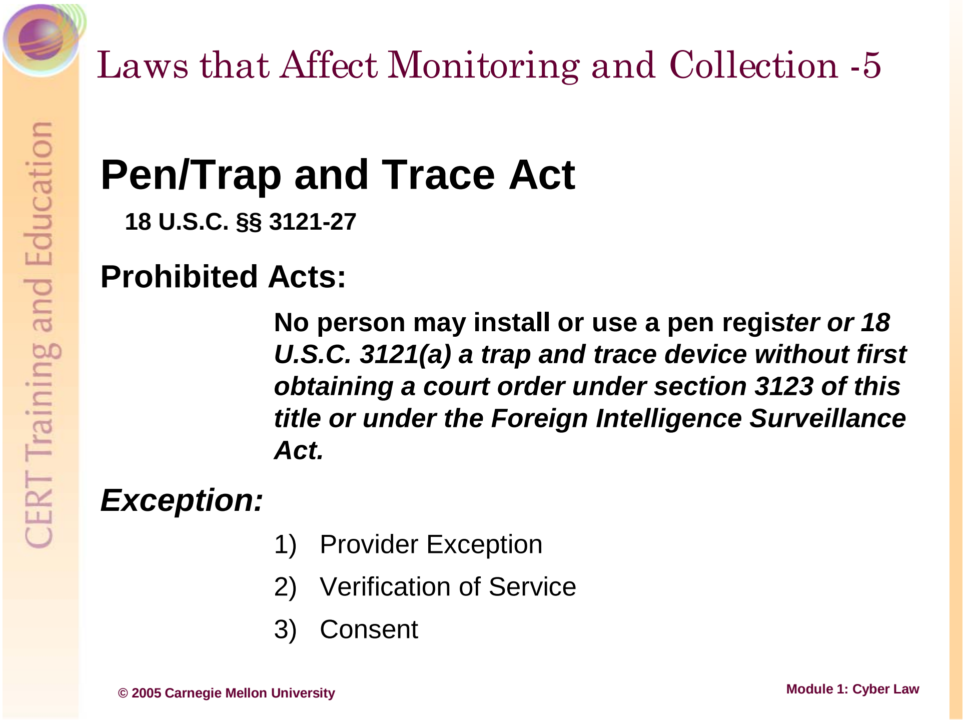
An example scenario of this exception would be as follows:

> A system administrator is on duty when she receives a remote syslog message from a Linux system that is running a SQL database indicating that the ftp service was started. Upon further inspection of the syslog files, she also discovers that moments before the ftp alert, a "root" login occurred. This clearly presents a security threat to that system. There is an unauthorized root login and potentially an ftp session in progress. At this point, in an effort to protect the network, the system administrator runs a sniffer to begin monitoring and collecting the entire session originating from that system.[*]

It is also a best practice to post banners on network devices. This practice was more important prior to the U.S. Patriot Act, when it was unclear whether a "trespasser" enjoyed protection under the Wiretap Act. With the Patriot Act, there is an exception that specifically deals with trespassers. However, posting banners on network devices indicating that monitoring is occurring and how that monitoring will be used is still a best practice.

[12] *Fraud and Related Activity in Connection with Computers* (see http://www.cybercrime.gov/1030_new.html)

[*] This example is not intended to be legally authoritative but to frame a legal concept in terms that a system administrator will understand.



CERT Training and Education

Laws that Affect Monitoring and Collection -5

Pen/Trap and Trace Act

18 U.S.C. §§ 3121-27

Prohibited Acts:

No person may install or use a pen regis*ter or 18 U.S.C. 3121(a) a trap and trace device without first obtaining a court order under section 3123 of this title or under the Foreign Intelligence Surveillance Act.*

Exception:

1) Provider Exception
2) Verification of Service
3) Consent

© 2005 Carnegie Mellon University

Module 1: Cyber Law

1.4.2.2 Pen Registers and Trap and Trace Devices

Whereas the Wiretap Act focuses on the content of electronic communications, this statute, 18 U.S.C. §§ 3121-27, prohibits the installation of a device that records or decodes dialing, routing, addressing, or signaling information for outgoing (Pen Registers) and incoming (Trap and Trace) wired or electronic communications, except for certain purposes.

Statutory definitions:

- Pen Register - 18 U.S.C.§ 3127(3)

  The term "pen register" means a device or process which records or decodes dialing, routing, addressing, or signaling information transmitted by an instrument or facility from which a wire or electronic communication is transmitted, provided, however, that such information shall not include the contents of any communication, but such term does not include any device or process used by a provider or customer of a wire or electronic communication service for billing, or recording as an incident to billing, for communications services provided by such provider or any device or process used by a provider or customer of a wire communication service for cost accounting or other like purposes in the ordinary course of its business.

- Trap and Trace -18 U.S.C.§ 3127(4)

  The term "trap and trace device" means a device or process which captures the incoming electronic or other impulses which identify the originating number or other dialing, routing, addressing, and signaling information reasonably likely to

> *identify the source of a wire or electronic communication, provided, however, that such information shall not include the contents of any communication.*

Within the definition one major point is clearly made: these devices (pen registers and trap and trace devices) deal with non-content information. This statute governs the use of tools that are used to record and collect signaling information (i.e., Layers 3 and 4), such as Netlog and Argus. Notice that there is an absence of intent-specific terms.

A prohibited act under this statute, 18 U.S.C. § 3121(a), specifies that

> *no person may install or use a pen register or a trap and trace device without first obtaining a court order…*

There are three exceptions to this statute, 18 U.S.C. §§§ 3121(b)(1), (2), and (3).

1) Provider Exception 18 U.S.C. § 3121(b)(1)

   > *…relating to the operation, maintenance, and testing of a wire or electronic communication service or to the protection of the rights or property of such provider, or to the protection of users of that service from abuse of service or unlawful use of service…*

   The following is an example scenario for this exception:

   > A network administrator runs Argus to monitor and analyze netflow traffic. Although this is a collection of signaling information, it is permitted under the Provider Exception because it is related to the maintenance and protection of the provider.*

2) Verification of Service 18 U.S.C. § 3121(b)(2)

   > *…to record the fact that a wire or electronic communication was initiated or completed in order to protect such provider, another provider furnishing service toward the completion of the wire communication, or a user of that service, from fraudulent, unlawful or abusive use of service...*

   The following is an example scenario for this exception:

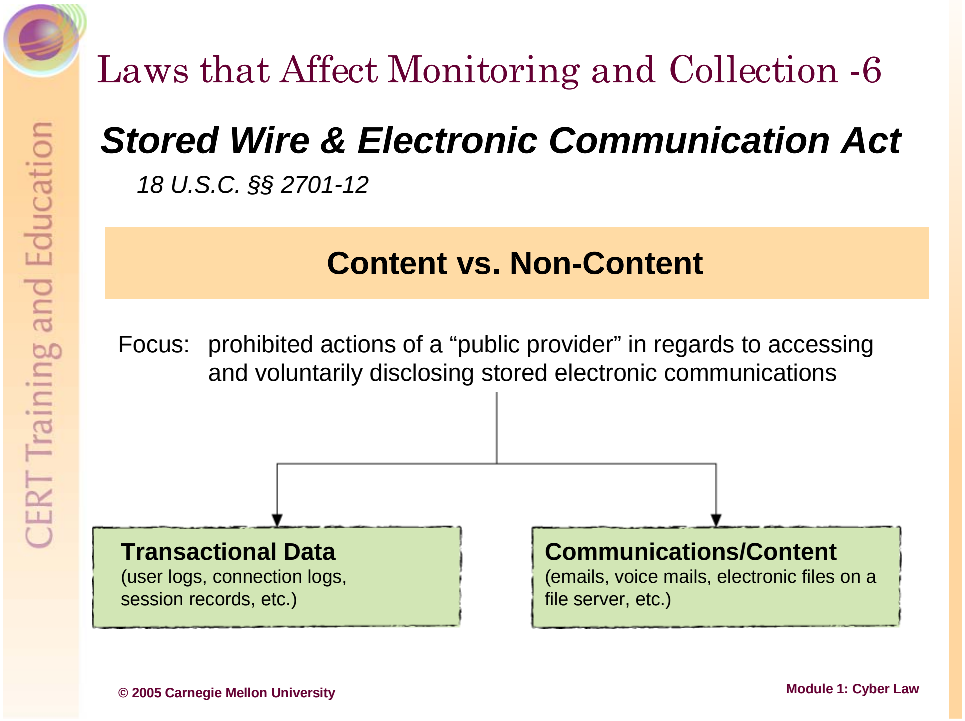
   > Companies that offer services on a per use basis maintain records of usage to ensure proper billing. For example, some cellular telephone companies offer text messaging on a per use basis. Records of the number and size of the messages sent to and from an account will likely be collected.*

[*] This example is not intended to be legally authoritative but to frame a legal concept in terms that a system administrator will understand.

3) Consent 18 U.S.C. § 3121(b)(3)

> …*where the consent of the user of that service has been obtained*.

Again, here is where bannering the network and written and signed acceptable use policies are critical. If you have a correctly worded banner and an acceptable use policy that informs users of monitoring activities and how the information gathered from monitoring will be used, in most cases the consent burden will be satisfied.



CERT Training and Education

Laws that Affect Monitoring and Collection -6

Stored Wire & Electronic Communication Act

18 U.S.C. §§ 2701-12

Content vs. Non-Content

Focus: prohibited actions of a "public provider" in regards to accessing and voluntarily disclosing stored electronic communications

Transactional Data
(user logs, connection logs, session records, etc.)

Communications/Content
(emails, voice mails, electronic files on a file server, etc.)

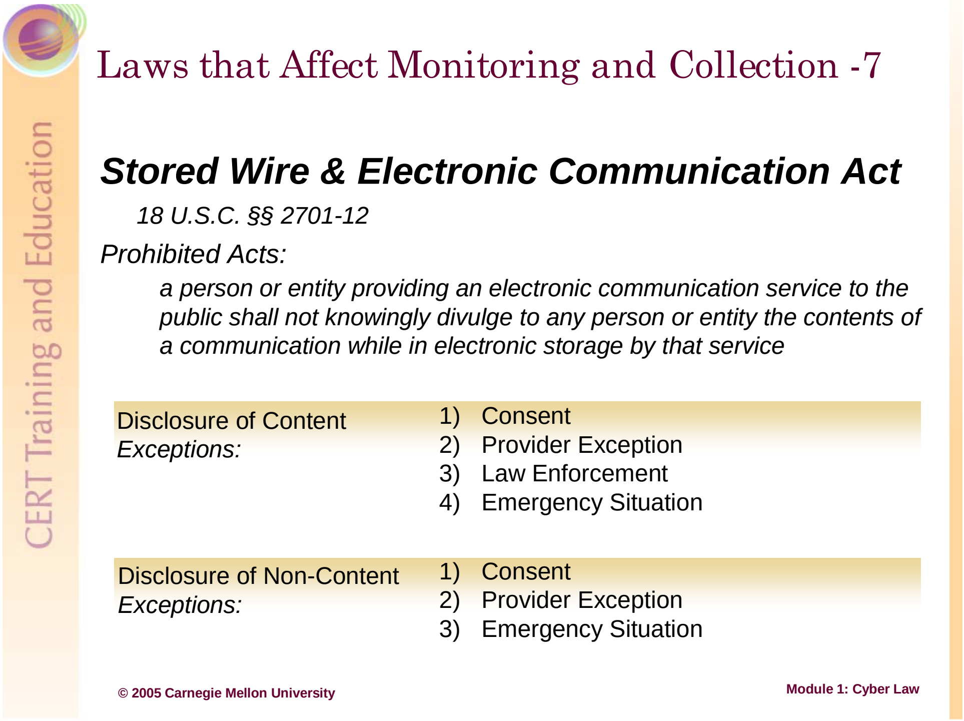
© 2005 Carnegie Mellon University

Module 1: Cyber Law

1.4.2.3 Stored Wired and Electronic Communications Act

The Stored Wired and Electronic Communication Act, 18 U.S.C. §§ 2701-12, deals with general protections for stored communications. There are several other statutes that may affect the level of protection, access, and disclosure of stored electronic communication, such as HIPAA and the Family Educational Rights and Privacy Act (FERPA). Consultation with legal counsel should be sought regarding whether your organization may be governed by these special types of privacy laws.

Like the Wiretap Act and the Pen/Trap and Trace Statute, the Stored Wired and Electronic Communication Act deals with content and non-content differently.



CERT Training and Education

Laws that Affect Monitoring and Collection -7

Stored Wire & Electronic Communication Act

18 U.S.C. §§ 2701-12

Prohibited Acts:

> *a person or entity providing an electronic communication service to the public shall not knowingly divulge to any person or entity the contents of a communication while in electronic storage by that service*

Disclosure of Content *Exceptions:*

1) Consent
2) Provider Exception
3) Law Enforcement
4) Emergency Situation

Disclosure of Non-Content *Exceptions:*

1) Consent
2) Provider Exception
3) Emergency Situation

© 2005 Carnegie Mellon University

Module 1: Cyber Law

Whereas the Wiretap and Pen/Trap and Trace statutes deal with real-time communications, this statute deals with and provides general protection from access to and disclosure of stored electronic communications. There are several sections to this statute, but for our purpose we will look only at the prohibited acts of *public providers* regarding access and voluntary disclosure of stored electronic communications. This section does not prohibit the actions of private providers who may voluntarily disclose content (stored voice or email), transactional data, and user information to government and non-government entities. Whether your organization is a public provider or private provider should be determined by counsel.

There are more prohibited acts under this statute than addressed here. This is a very complex statute that contains various restrictions. For the purpose at hand, only the following prohibited acts under the statute (18 U.S.C. § 2702) will be discussed:

> *(1) a person or entity providing an electronic communication service to the public shall not knowingly divulge to any person or entity the contents of a communication while in electronic storage by that service; and*
>
> *(2) a person or entity providing remote computing service to the public shall not knowingly divulge to any person or entity the contents of any communication which is carried or maintained on that service—*
>
> *(3) a provider of remote computing service or electronic communication service to the public shall not knowingly divulge a record or other information pertaining to a subscriber to or customer of such service (not including the*

> *contents of communications covered by paragraph (1) or (2)) to any governmental entity.*

This statute breaks down stored communication into three areas: (1) communications content (emails, voice mails, electronic files on a file server, etc.), (2) transactional data (user logs, connection logs, session records, etc.), and (3) subscriber information. This breakdown has more to do with the mandatory disclosure section of this statute than with issues that system administrators and network security personnel will deal with, but is nevertheless worth noting.

The exceptions to this statute are broken into two categories: content (stored emails, electronic files, etc.) and non-content (subscriber and transactional data related to account activities). There is some overlap between these two areas. The following are exceptions that relate to the potential actions of system administrators and network security personnel.

Exceptions for Disclosure of Communications Content

> *A person or entity may divulge the contents of a communication…*

1) Consent 18 U.S.C. § 2702(b)(3)

   > *with the lawful consent of the originator or an addressee or intended recipient of such communication, or the subscriber in the case of remote computing service…*

   > For example, some companies prohibit the personal use of company email. To enforce this policy, consent from the users is required to allow for monitoring of email. Given this consent, preferably with both a written and signed policy and a banner, compliance checks are possible.[*]

   Consent should never be taken as a general permission for system administrators to conduct ad hoc, random monitoring activities. Rather, monitoring activities related to consent should be part of a well-documented process.

2) Provider Exception 18 U.S.C. § 2702(b)(5)

   > *as may be necessarily incident to the rendition of the service or to the protection of the rights or property of the provider of that service.*

   Products such as email virus scanners may very well fall under this definition, if an application is scanning the contents of stored communications (email) on the mail server before sending it or prior to delivering it. Again, there is a clear nexus between what is being monitored and the threat.[*]

[*] This example is not intended to be legally authoritative but to frame a legal concept in terms that a system administrator will understand.

3) Law Enforcement 18 U.S.C. §§ 2702(b)(6)(A)(i)and(ii)

> *…if the contents were inadvertently obtained by the service provider and appear to pertain to the commission of a crime…*

> For example, in the examination of email for compliance with company policy, a document is discovered that contains questionable material. The important issue here is that the discovery was incidental, the result of a regular process.*

4) Emergency Situations 18 U.S.C. § 2702(b)(6)(C)

> *…if the provider reasonably believes that an emergency involving immediate danger of death or serious physical injury to any person requires disclosure of the information without delay…*

There are numerous scenarios in which this exception applies. It is often involved in suicide notes, missing children, etc.

Exceptions for Disclosure of Non-Content Customer Records

> *…may divulge a record or other information pertaining to a subscriber to or customer of such service (not including the contents of communications covered by subsection…*

1) Consent 18 U.S.C. § 2702(c)(2)

> *with the lawful consent of the customer or subscriber…*

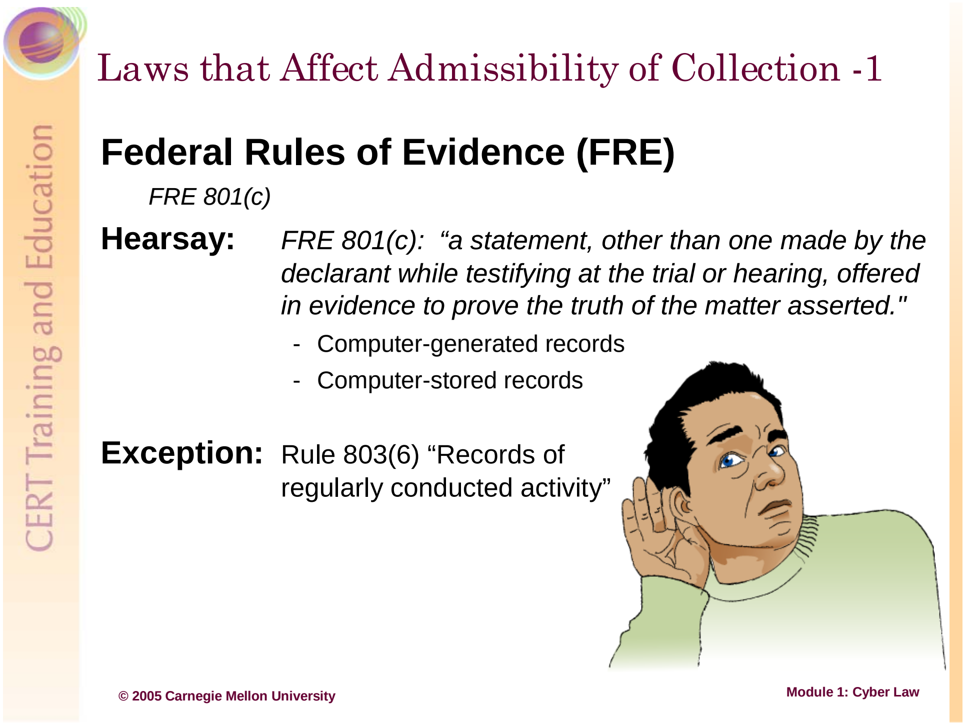
2) Provider Exception 18 U.S.C. § 2702(c)(3)

> *as may be necessarily incident to the rendition of the service or to the protection of the rights or property of the provider of that service…*

3) Emergency Situations 18 U.S.C. § 2702(c)(4)

> *to a governmental entity, if the provider reasonably believes that an emergency involving immediate danger of death or serious physical injury to any person justifies disclosure of the information…*

There are numerous scenarios in which this exception applies. It is often involved in suicide notes, missing children, etc.



Laws that Affect Admissibility of Collection -1

Federal Rules of Evidence (FRE)

FRE 801(c)

Hearsay: *FRE 801(c): "a statement, other than one made by the declarant while testifying at the trial or hearing, offered in evidence to prove the truth of the matter asserted."*

- Computer-generated records
- Computer-stored records

Exception: Rule 803(6) "Records of regularly conducted activity"

© 2005 Carnegie Mellon University

Module 1: Cyber Law

1.5 Legal Governance Related to Admissibility (Federal Rules of Evidence)

1.5.1 Hearsay

Hearsay is defined by the Federal Rule of Evidence 801(c) as "a statement, other than one made by the declarant while testifying at the trial or hearing, offered in evidence to prove the truth of the matter asserted." Further, according to Rule 802, "Hearsay is not admissible except as provided by these rules or by other rules prescribed by the Supreme Court pursuant to statutory authority or by Act of Congress."

Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations, published by the Computer Crime and Intellectual Property Section, Criminal Division, of the United States Department of Justice, is an excellent source for detailed information related to this topic, as well as other general topics dealing with electronic evidence [USDOJ 02]. Its importance to this topic relates to understanding the difference between computer-generated records and computer-stored records. Basically there are two types of computer records: those that "contain the writings of some person or persons and happen to be in electronic form" (i.e., computer-stored records) and those that "contain the output of computer programs, untouched by human hands" (i.e., computer-generated records).[13]

It is important to understand the difference between these two types of computer records because computer-stored records can contain hearsay and computer-generated records, which IT professionals are more likely to deal with, cannot. In practice, however, most federal courts have viewed all computer records as potentially containing hearsay. But this view is likely to change:

> *As the federal courts develop a more nuanced appreciation of the distinctions to be made between different kinds of computer records, they are likely to see that the admission of computer records generally raises two distinct issues. First, the government must establish the authenticity of all computer records by providing "evidence sufficient to support a finding that the matter in question is what its proponent claims." Fed. R. Evid. 901(a). Second, if the computer records are computer-stored records that contain human statements, the government must show that those human statements are not inadmissible hearsay [Kerr 01].*

In dealing with computer-generated records, two issues arise: authentication and reliability, which will be explained in subsequent sections.

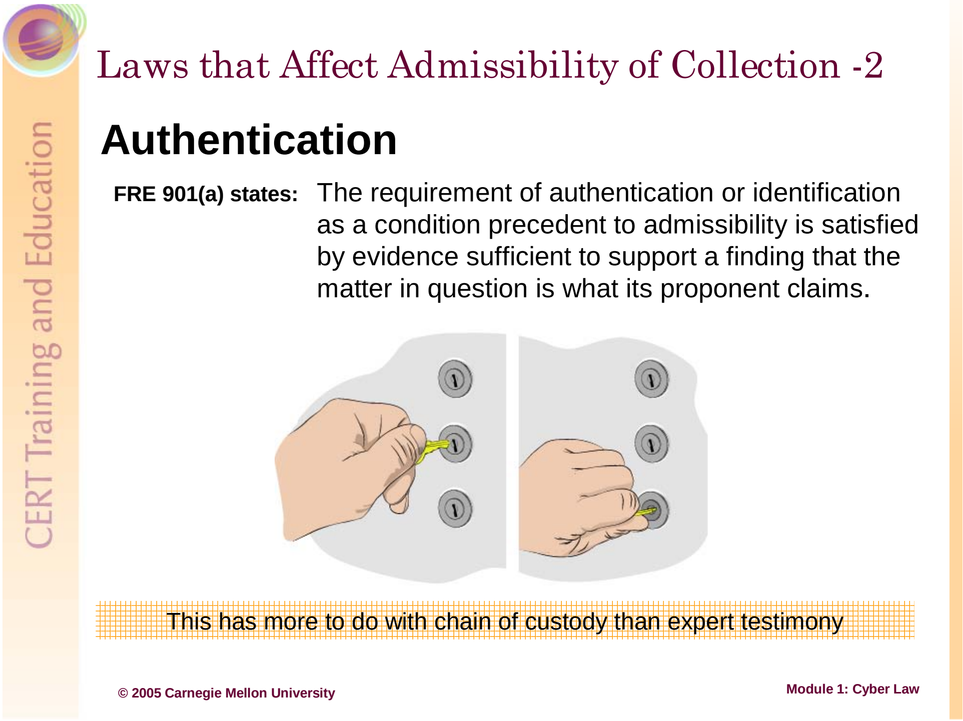
1.5.1.1 Exceptions

There are twenty-four enumerated exceptions to the Hearsay Rule. Rule 803(6), "records of regularly conducted activity," deals with issues pertinent to computer forensics. This exception deals with "regularly conducted business activities." This is why an IT policy is so critical regarding network monitoring and incident response. The outcomes from random, ad hoc procedures will not fall under this exception. Rule 803(6) reads as follows:

> *A memorandum, report, record, or data compilation, in any form, of acts, events, conditions, opinions, or diagnoses, made at or near the time by, or from information transmitted by, a person with knowledge, if kept in the course of a regularly conducted business activity, and if it was the regular practice of that business activity to make the memorandum, report, record, or data compilation, all as shown by the testimony of the custodian or other qualified witness, unless the source of information or the method or circumstances of preparation indicate lack of trustworthiness. The term "business" as used in this paragraph includes business, institution, association, profession, occupation, and calling of every kind, whether or not conducted for profit.*

The best example of this would be logging. If a company logs everything as a practice, it is very likely that those logs will be admissible. However, if only partial logging is done and during an incident additional logging is turned on, a question of admissibility may arise.

[13] In some cases, records are both computer generated and computer stored. A spreadsheet, for example, could contain both data entered by a person and data generated through calculations done by macros [Kerr 01].



CERT Training and Education

Laws that Affect Admissibility of Collection -2

Authentication

FRE 901(a) states: The requirement of authentication or identification as a condition precedent to admissibility is satisfied by evidence sufficient to support a finding that the matter in question is what its proponent claims.

This has more to do with chain of custody than expert testimony

© 2005 Carnegie Mellon University

Module 1: Cyber Law

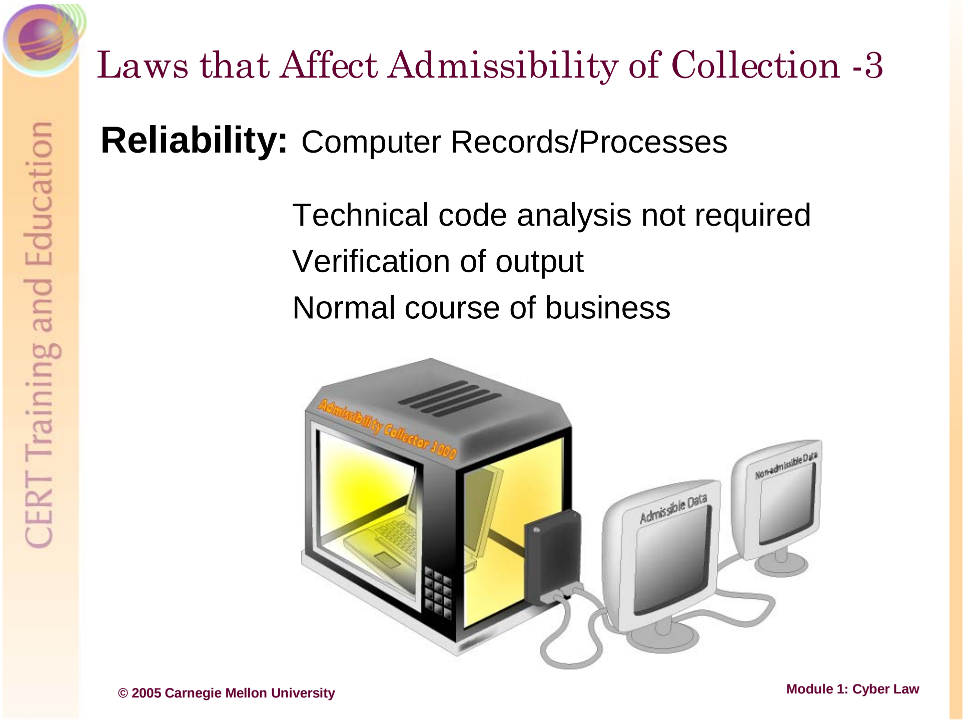
1.5.2 Authentication

It is the burden of the person/party attempting to admit a computer-generated record to authenticate that record. Federal Rule of Evidence 901(a) states the following:

> *The requirement of authentication or identification as a condition precedent to admissibility is satisfied by evidence sufficient to support a finding that the matter in question is what its proponent claims.*

To authenticate a computer record, a person who acquired the record or caused the record to be generated can testify to the authenticity of the record (i.e., say that the record is what it purports to be). An example of this is when a system administrator records the open connections on a system by running a *netstat* command and has that information saved on a diskette. This same system administrator would need to authenticate this diskette at court by testifying that he or she ran the command that caused the output and, further, that the output was saved to that diskette. It is not important or required that the system administrator have programming knowledge and understand how the *netstat* command works or to be able to provide analysis of the results of the *netstat* command. All that is required is that the system administrator has first-hand knowledge, i.e., that he or she ran the command and generated the output that was then saved to the diskette.

In addition, *chain of custody* plays a significant role in the process of authentication. It is not enough to just testify after the fact about what was collected. Having a process in place that will track collected information and ensure that it is preserved and not tampered with is also required.



CERT Training and Education

Laws that Affect Admissibility of Collection -3

Reliability: Computer Records/Processes

Technical code analysis not required

Verification of output

Normal course of business

© 2005 Carnegie Mellon University

Module 1: Cyber Law

1.5.3 Reliability

Another issue that IT personnel should be prepared to deal with is the reliability of the computer programs that generate the computer records. This area is of particular interest to IT staff, since it is where software that is used in the collection of information related to an incident may be challenged.

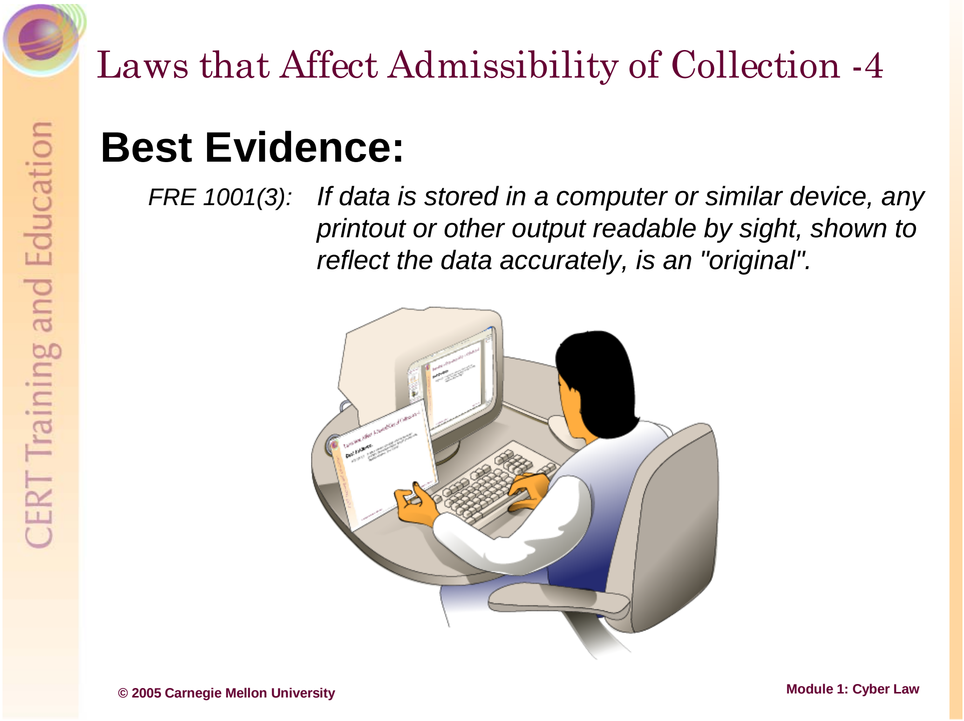
For the purpose of establishing reliability, computer programs are distinguished by whether they are or are not run in the normal course of business. An example of programs run in the normal course of business is RADIUS servers at an ISP supporting dial-up users. RADIUS servers provide several functions related to an ISP's business operations. For example, a RADIUS server would be used to maintain logs of when users dial in to access the network. The ISP normally uses this information for billing purposes, but the same information may become important in the event of an incident. When the information is used to support an incident, the fact the program that generated the information was used in the normal course of business gives it reliability [Kerr 01].

The more difficult situation in terms of reliability is when programs are used to collect information that are not run in the normal course of business. The guidance from the courts is more complicated in this area, as indicated in this excerpt from *Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations*:

> *The courts have indicated that the government can overcome this challenge so long as "the government provides sufficient facts to warrant a finding that the records are trustworthy and the opposing party is afforded an opportunity to*

> *inquire into the accuracy thereof[.]" United States v. Briscoe, 896 F.2d 1476, 1494 (7th Cir. 1990). See also Liebert, 519 F.2d at 547; DeGeorgia, 420 F.2d. at 893 n.11. Compare Fed. R. Evid. 901(b)(9) (indicating that matters created according to a process or system can be authenticated with "[e]vidence describing a process or system used…and showing that the process or system produces an accurate result")…the government may need to disclose "what operations the computer had been instructed to perform [as well as] the precise instruction that had been given" if the opposing party requests. Dioguardi, 428 F.2d at 1038 [USDOJ 02].*

How does this translate into working knowledge for IT staff? Module 2, "File Systems, Duplication, Wiping, and Trusted Tools," will describe some of the technical implications of dealing with reliability issues.



CERT Training and Education

Laws that Affect Admissibility of Collection -4

Best Evidence:

FRE 1001(3): *If data is stored in a computer or similar device, any printout or other output readable by sight, shown to reflect the data accurately, is an "original".*

© 2005 Carnegie Mellon University

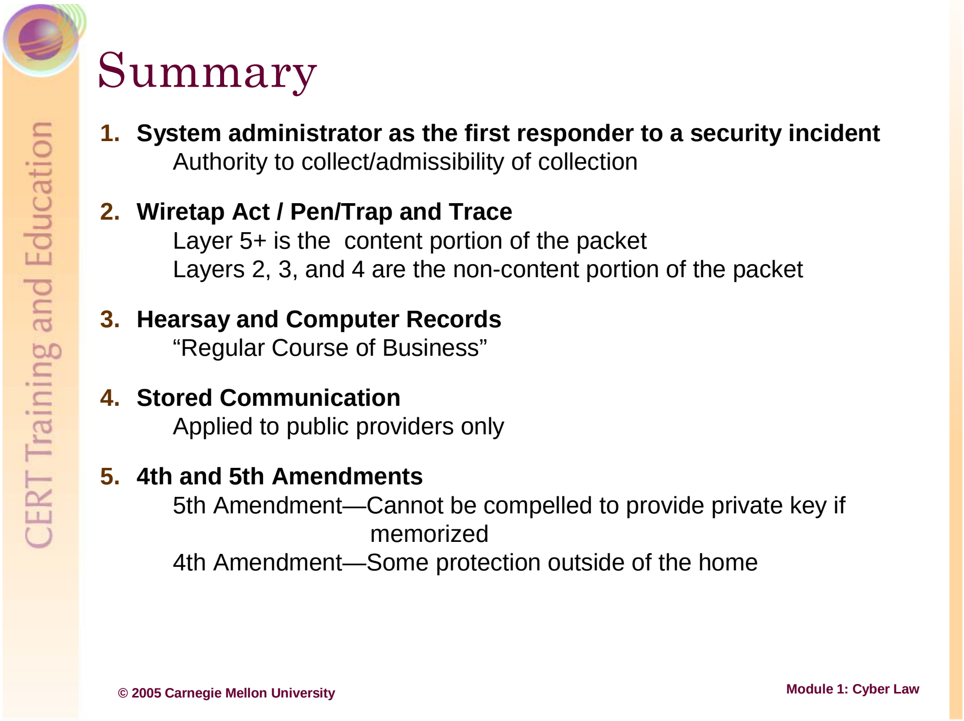
Module 1: Cyber Law

1.5.4 The Best Evidence Rule

The Best Evidence rule (Federal Rule of Evidence 1002) states that to prove the content of writing, recording, or photograph, the original writing, recording, or photograph is ordinarily required. Federal Rule of Evidence 1001(3) defines “original”:

> *If data are stored in a computer or similar device, any printout or other output readable by sight, shown to reflect the data accurately, is an “original”.*

The courts have long recognized that at times it may be both impractical and unreasonable to require originals, which might be entire computers or large volumes of data, for information that otherwise could be represented in a chart or graph. In fact, in the 9th Circuit the courts require that during seizures of computers, on-site imaging must be done, leaving the original behind and taking only a “copy,” that is, a true bit by bit forensic image.



CERT Training and Education

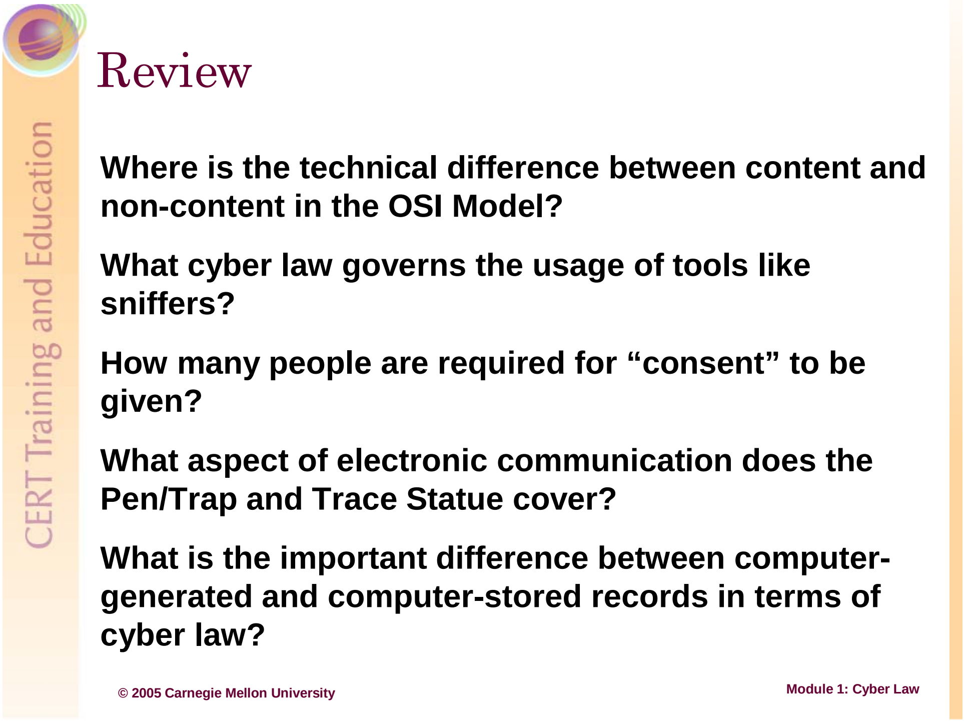
Summary

1. **System administrator as the first responder to a security incident**
   Authority to collect/admissibility of collection
2. **Wiretap Act / Pen/Trap and Trace**
   Layer 5+ is the content portion of the packet
   Layers 2, 3, and 4 are the non-content portion of the packet
3. **Hearsay and Computer Records**
   “Regular Course of Business”
4. **Stored Communication**
   Applied to public providers only
5. **4th and 5th Amendments**
   5th Amendment—Cannot be compelled to provide private key if memorized
   4th Amendment—Some protection outside of the home

Module 1: Cyber Law

1.6 Summary

The five points above are the key concepts that should be understood from this module.



CERT Training and Education

Review

Where is the technical difference between content and non-content in the OSI Model?

What cyber law governs the usage of tools like sniffers?

How many people are required for “consent” to be given?

What aspect of electronic communication does the Pen/Trap and Trace Statue cover?

What is the important difference between computer-generated and computer-stored records in terms of cyber law?

© 2005 Carnegie Mellon University

Module 1: Cyber Law

1.7 Review

Q. Where is the technical difference between content and non-content in the OSI Model?

A. Between Layers 4 and 5.

Q. What cyber law governs the usage of tools like sniffers?

A. The Wiretap Act.

Q. How many people are required to give “consent” under federal law?

A. Only one.

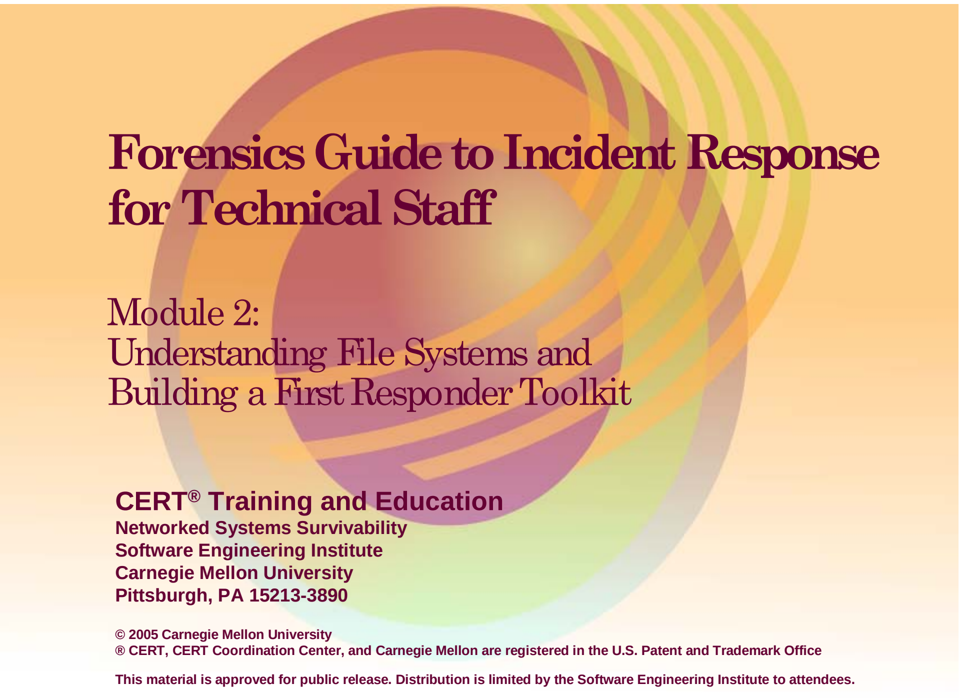
Q. What aspect of electronic communication does the Pen/Trap and Trace Statute cover?

A. The non-content portion of real-time communications.

Q. What is the important difference between computer-generated and computer-stored records in terms of cyber law?

A. Computer-generated records contain no human statements and computer-stored records may contain human statements.

2 Module 2: Understanding File Systems and Building a First Responder Toolkit



Forensics Guide to Incident Response for Technical Staff

Module 2:
Understanding File Systems and
Building a First Responder Toolkit

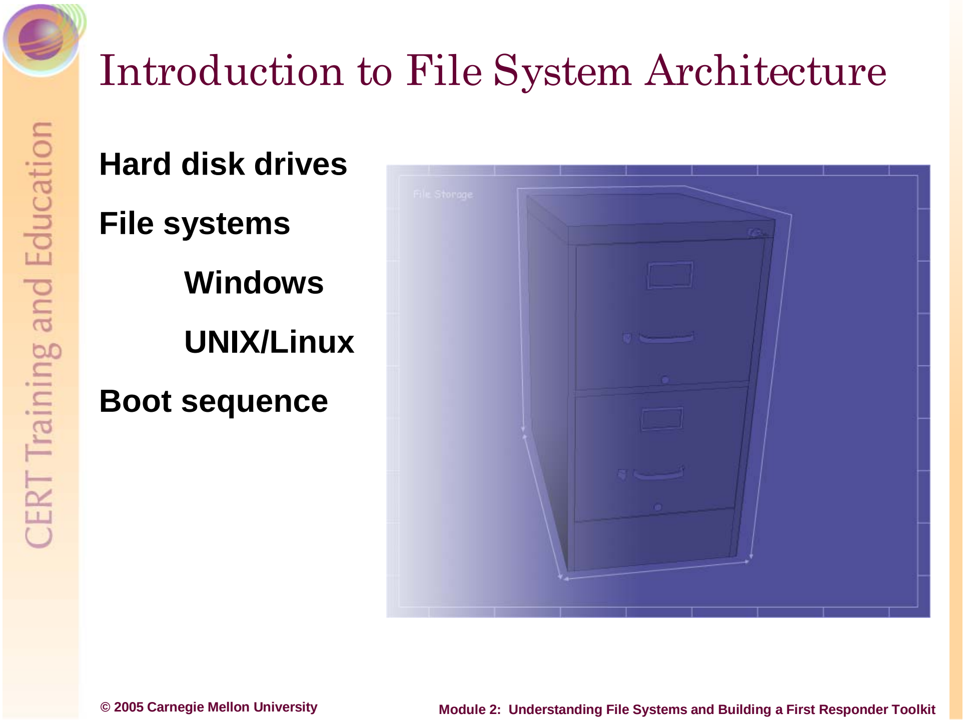
CERT® Training and Education
Networked Systems Survivability
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

2.1 Introduction

Module 2 consists of two distinct sections: Understanding File Systems, and Methodologies for Building a First Responder Toolkit.

The first section, Understanding File Systems, is an introduction to technical concepts and terms important for later modules. Its intent is to provide the novice with a foundation and the knowledgeable with a baseline cursory review. The first portion provides an overview of how the operating system (OS) interacts with physical storage media in creating and modifying files. Most storage occurs on a computer's internal hard drive and a basic understanding of hard drive design provides insight into the advantages and limitations of file systems. A cursory look at the similarities between Windows and Linux is highlighted with a brief discussion of each OS's characteristics. The last portion of this section addresses common forensic nomenclature.

The second section outlines a best practice methodology for creating a trusted first responder tool kit that system and network administrators can use when responding to security events or system anomalies on live systems. The goal of this step-by-step, repeatable, and well-documented approach is to ensure all information is collected using sound forensic practices.



CERT Training and Education

Introduction to File System Architecture

Hard disk drives

File systems

- Windows
- UNIX/Linux

Boot sequence

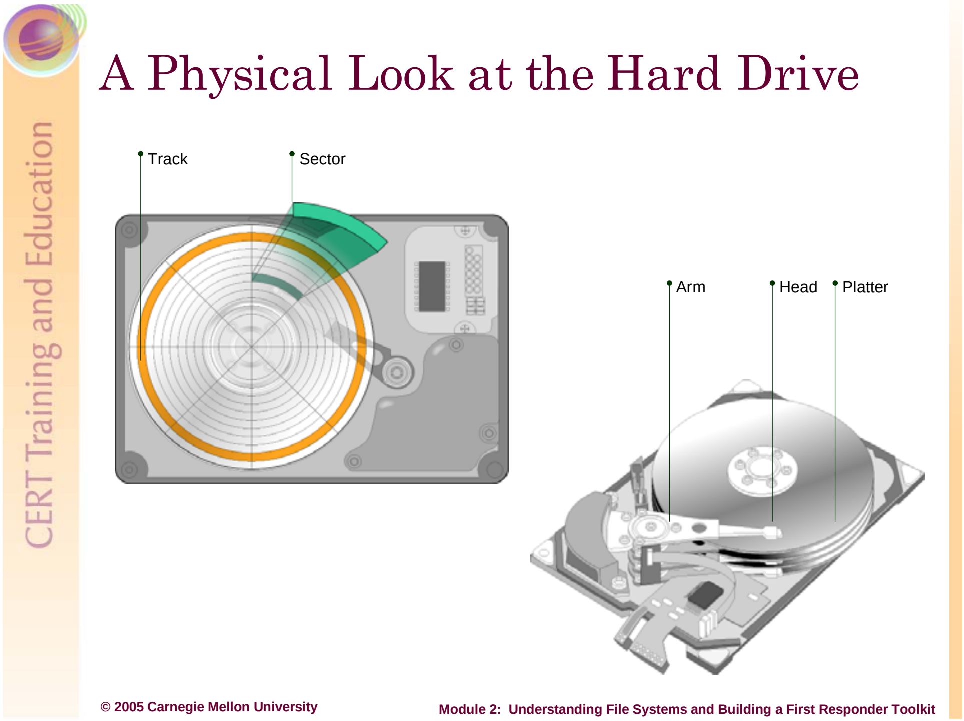
File Storage

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2 File System Architecture

File system design concepts and technical details fill up large volumes of textbooks, so a detailed discussion on this topic is well beyond the scope of this introductory handbook. This section will provide a basic understanding of the most popular storage media, the hard drive, and how its design interacts with an OS's file system. This overview of the Windows and Linux file systems will provide the foundation for the next two modules dealing with collection. Additionally, the vocabulary covered in the next few pages contributes to the technical staff's communication with a professional cyber investigator.

Because of the large array of different file systems, especially with regard to open-source OSs like Linux, you are encouraged to expand your knowledge outside the boundaries of this handbook for your particular networks. An understanding of what is expected in the normal use of a file system (where typically certain files are placed, what default directories are named, etc.) assists in the ability to notice anomalies – which is the first step toward incident detection.



CERT Training and Education

A Physical Look at the Hard Drive

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.1 Physical Look at the Hard Drive

To understand how file systems work on different OSs, you must be familiar with the physical organization of the hard disk drive. A hard drive is like a record player, except that the hard drive can handle multiple stacks of records. On a hard drive, the magnetic media are called platters, and data is written on them by a read/write arm that has a head (analogous to a needle).

Hard drives consist of several platters with heads inserted between them that can read on one or both sides. The data on each platter is physically organized into tracks and sectors. A track is an individual concentric circle on the platter. A combination of tracks is called a cylinder. A sector divides a track and typically holds 512 bytes.

By knowing the attributes of a hard disk drive, you can determine its maximum storage capacity.

Bytes = C/D * H/C * S/T * 512 bytes

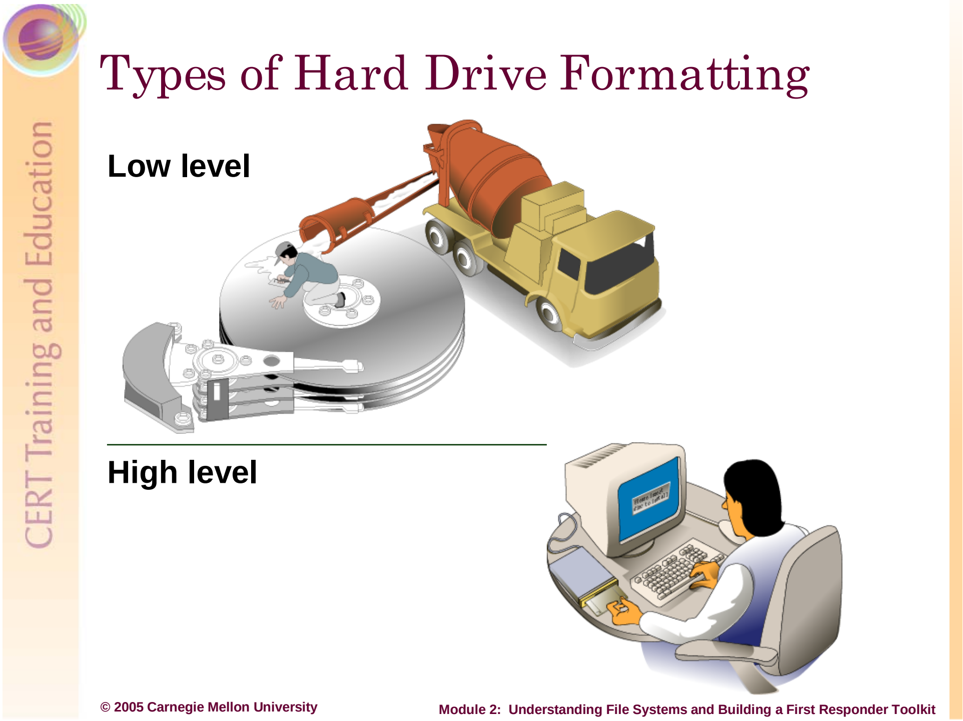
C is the number of cylinders

T is the number of tracks (track numbering starts with 0, not 1)

S is the number of sectors

For more details, visit the following Web sites:

- http://www.pcguide.com/ref/hdd/
- http://www.howstuffworks.com/hard-disk.htm



CERT Training and Education

Types of Hard Drive Formatting

Low level

High level

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

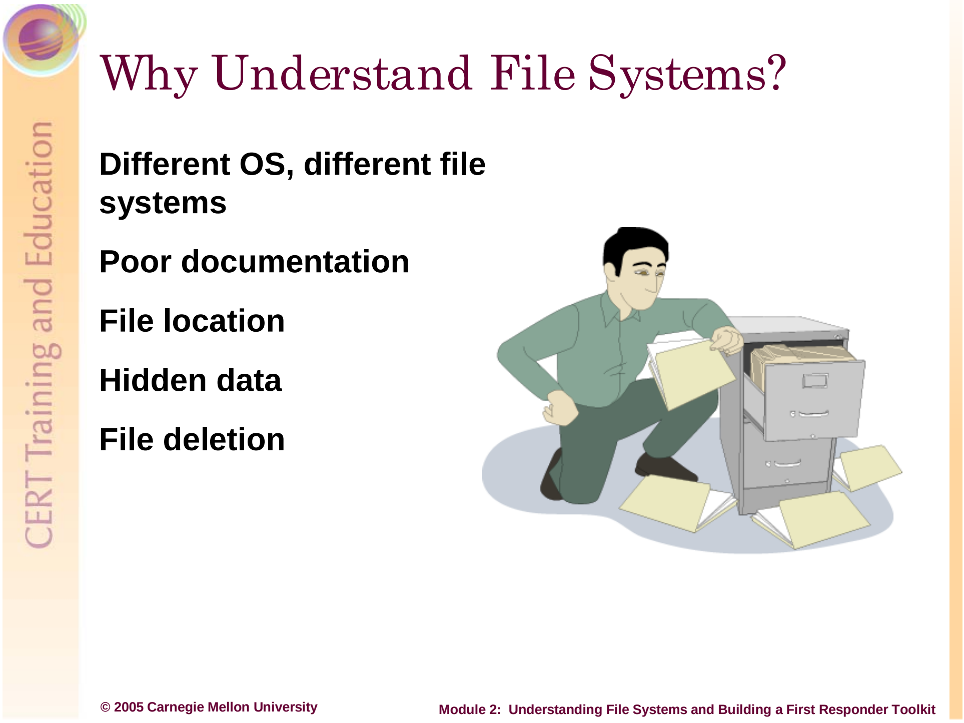
2.2.2 Types of Hard Drive Formatting

The two types of hard drive formatting are low and high level. Low-level formatting creates the track and sectors on the drive. These tracks and sectors form the physical blocks of storage of 512 bytes each.

High-level formatting is file system specific, including Microsoft (DOS, FAT, and NTFS) and Open Source Initiative (OSI) varieties (ext2, ReiserFS, and XFS).

Low-level formatting is done by the hard drive vendor. High-level formatting is done when the OS is installed.

High-level formatting creates the hard drive's file system and allows the OS to store files by dividing them into smaller pieces and saving them in separate clusters (a grouping of sectors) on the disk. The OS uses this file system to keep track of the placement and sequence of each piece and to identify which sectors on the disk are free and available for new files. The computer can then assemble the different pieces when a file is viewed or executed.



CERT Training and Education

Why Understand File Systems?

- Different OS, different file systems
- Poor documentation
- File location
- Hidden data
- File deletion

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.3 Importance of File Systems

A file system has two basic functions that impact the computer’s performance:

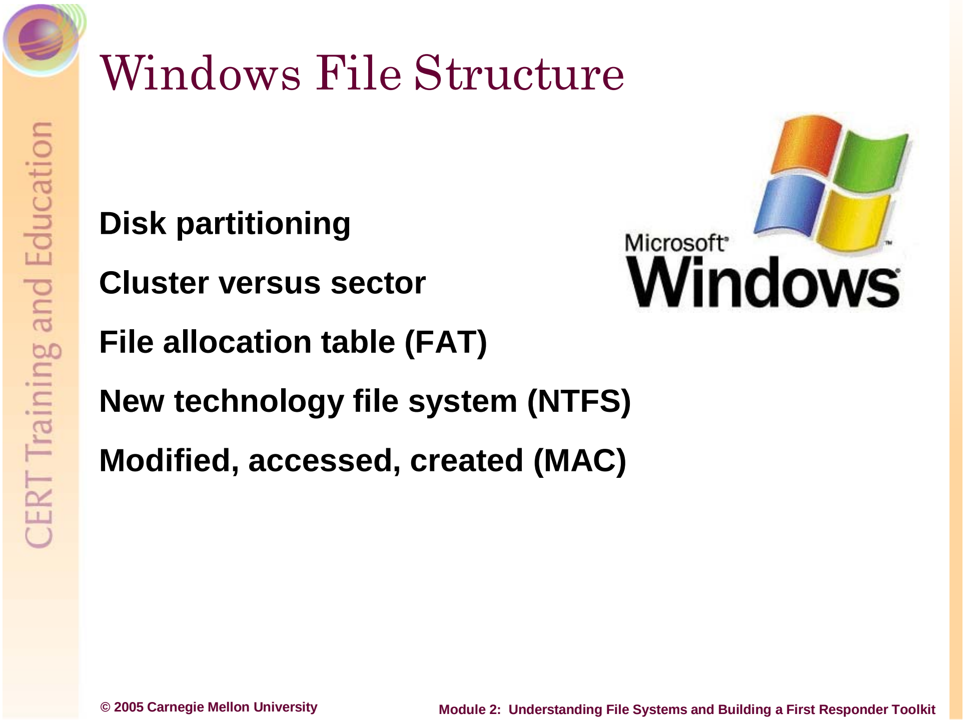
- mapping physical spaces on the drive to logical addresses that comprise files
- read/write capability to open, change, and delete files

Understanding how these functions work on different file systems is the foundation for responding to a security incident. Most file systems are related directly to a particular OS, although some OSs combine file systems.

To discover where files are located and how they are distributed, you need to know how to access and modify system settings when necessary. This is especially important because files can be hidden.

Typically when a user deletes a file, the file system does not permanently erase (wipe) the file from the hard drive. It simply creates a flag that tells the OS that the sector can be reused. Knowing how to rebuild files from the file system is one of the most important skills of the forensic examiner.

The forensic examiner has access to deleted files and to files contained in swap space, which is part of the virtual memory created on the hard drive by the OS. Swap space files are described later in this module.


CERT Training and Education
Microsoft Windows

Windows File Structure

- Disk partitioning
- Cluster versus sector
- File allocation table (FAT)
- New technology file system (NTFS)
- Modified, accessed, created (MAC)

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.4 Understanding Windows File Structure

On a physical hard disk, more then one logical section may exist. Dividing a physical drive into separate pieces is called partitioning. The partitions of logical drives are traditionally labeled by the file system as C, D, E, and so on.

Microsoft Windows is the predominate OS for user workstations, and has gained significant momentum in server applications. To understand Microsoft's file structure, we must start by defining some storage terms: clusters, file allocation tables (FAT), and the new technology file system (NTFS).

The hard drive's sectors, as previously described, are further grouped by the Microsoft file system into clusters. Clusters contain groups of sectors. These clusters then form larger data groups to make a single larger addressable storage unit. By combining sectors into clusters, the file system reduces the overhead to write and read files to the disk as it has to keep track of fewer unique storage areas. Usually the number of sectors to a cluster can be seen below for the Windows XP OS. The following excerpt is from the Microsoft Windows XP Professional Resource Kit Documentation [Microsoft 04].

> *A* cluster *(or* allocation unit*) is the smallest amount of disk space that can be allocated to hold a file. All file systems used by Windows XP Professional organize hard disks based on cluster size, which is determined by the number of sectors that the cluster contains. For example, on a disk that uses 512-byte sectors, a 512-byte cluster contains one sector, whereas a 4-KB cluster contains eight sectors.*

The two primary Windows file systems are FAT and NTFS. Both will be described later.

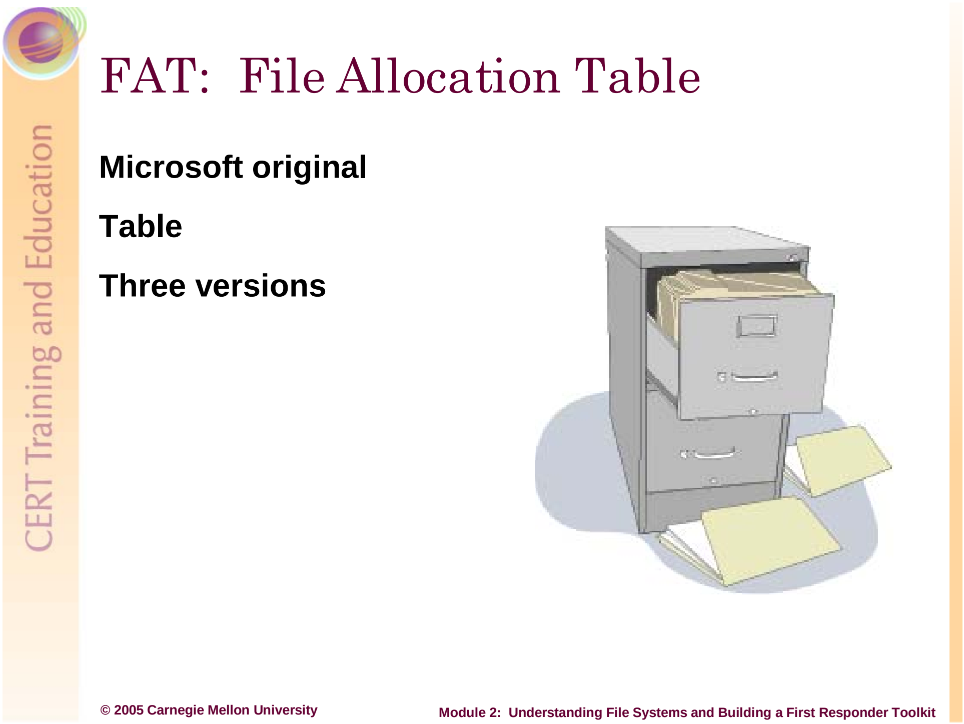
FAT16, FAT32, and NTFS each use different cluster sizes depending on the size of the volume, and each file system has a maximum number of clusters it can support. When the cluster size is smaller, the disk stores information more efficiently because unused space within a cluster cannot be used by other files. And when more clusters are supported, you can create and format larger volumes by using a particular file system.

Table 1 compares FAT16, FAT32, and NTFS volume and default cluster sizes.

Table 1: Default Cluster Sizes for Volumes with Windows XP Professional File Systems

Volume Size	**FAT16 Cluster Size**	**FAT32 Cluster Size**	**NTFS Cluster Size**
7 MB–16 MB	2 KB	Not supported	512 bytes
17 MB–32 MB	512 bytes	Not supported	512 bytes
33 MB–64 MB	1 KB	512 bytes	512 bytes
65 MB–128 MB	2 KB	1 KB	512 bytes
129 MB–256 MB	4 KB	2 KB	512 bytes
257 MB–512 MB	8 KB	4 KB	512 bytes
513 MB–1,024 MB	16 KB	4 KB	1 KB
1,025 MB–2 GB	32 KB	4 KB	2 KB
2 GB–4 GB	64 KB	4 KB	4 KB
4 GB–8 GB	Not supported	4 KB	4 KB
8 GB–16 GB	Not supported	8 KB	4 KB
16 GB–32 GB	Not supported	16 KB	4 KB
32 GB–2 terabytes	Not supported	Not supported	4 KB

A very helpful feature of Windows file systems is the details concerning MAC times, which stands for modified, accessed, and created. Date and time properties are appended to files as they are created, accessed, and modified. Examine a file’s MAC times for additional information about the history of the file, including a user’s interaction with it.



CERT Training and Education

FAT: File Allocation Table

Microsoft original

Table

Three versions

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.5 FAT: File Allocation Table

The following is from Microsoft’s Intellectual Property and Licensing Group introduction to FAT [Microsoft 03a]:

> *The first FAT file system was developed by Microsoft in 1976. That system was based on the BASIC programming language and allowed programs and data to be stored on a floppy disk. Since that time, the FAT file system has been improved upon multiple times to take advantage of advances in computer technology, and to further refine and enrich the FAT file system itself.*
>
> *Today, the FAT file system has become the ubiquitous format used for interchange of media between computers, and, since the advent of inexpensive, removable flash memory, also between digital devices. The FAT file system is now supported by a wide variety of OSs running on all sizes of computers, from servers to personal digital assistants. In addition, many digital devices such as still and video cameras, audio recorders, video game systems, scanners, and printers make use of FAT file system technology.*

The FAT database table contains the filenames, directory names, MAC times, cluster number, and assorted attributes (hidden, read-only, etc.) of disk’s files. This table resides at the outermost portion of the disk with FAT32 offering a redundant copy in the event the primary suffers damage. The recovery of the redundant table is crucial if malicious activity makes the primary table unreadable.

For a history of the FAT file system, visit the Microsoft Web site [Microsoft 03b].

Figure 2 [Stoffregen 03] shows the overall logical layout of the FAT32 file system, with the allocation tables, primary and backup (labeled FAT #1 and #2 respectively), written directly behind the volume ID and boot sector (contained in the reserved sectors).

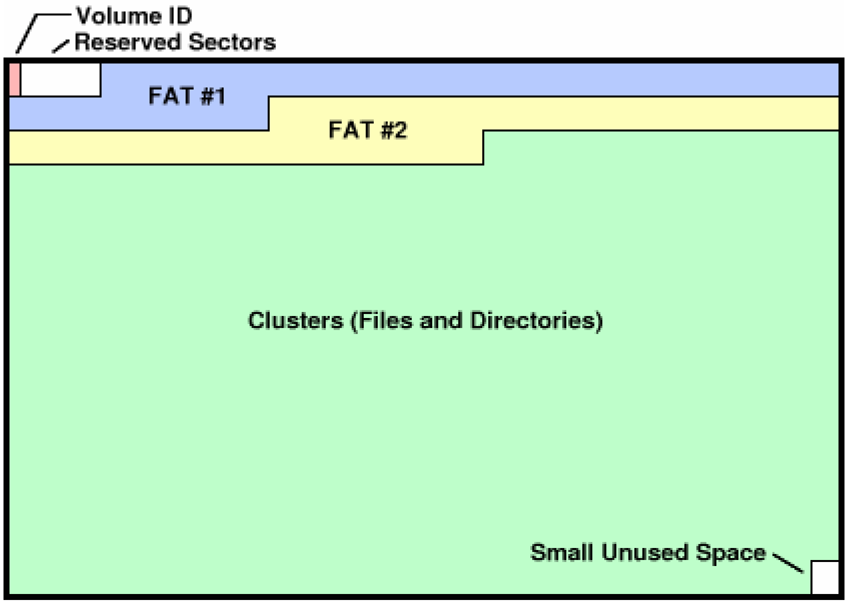
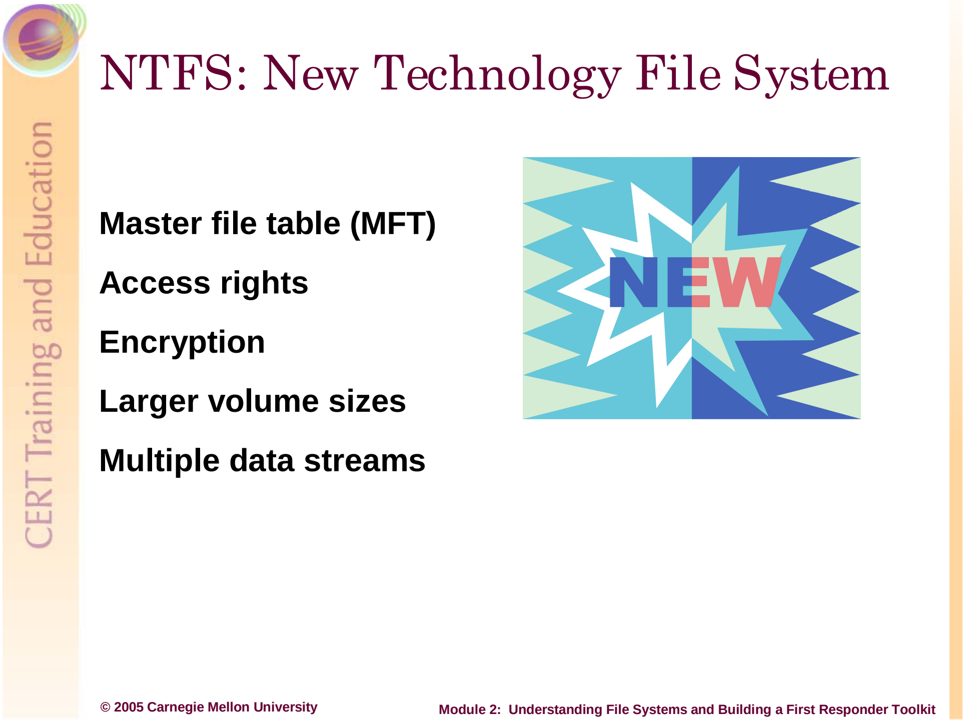


Figure 2: Logical Layout of the FAT32 File System



CERT Training and Education

NTFS: New Technology File System

- Master file table (MFT)
- Access rights
- Encryption
- Larger volume sizes
- Multiple data streams

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.6 NTFS: New Technology File System

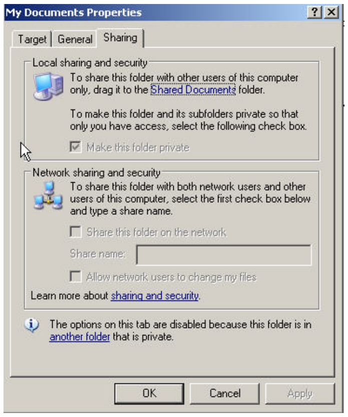
NTFS is currently the primary file system used by Windows XP. It was first introduced with Windows NT.

The Microsoft Windows XP Professional Resource Kit Documentation [Microsoft 04] is a comprehensive NTFS resource. The following excerpts are from the site.

> *NTFS replaces the FAT and uses a master file table (MFT), which is the first file on the disk. Records within the MFT are called meta-data and this contains information on all files located on the disk, including system files. A key advancement is the way files and directories are both stored on the disk with attributes that include security information. At format the MFT assigns logical cluster numbers (LCN) to the disk's entire partition. These LCNs allow the OS to read and write data on the disk. Each LCN is similarly linked to a virtual cluster number (VCN) which allows files to extend beyond across the free disk space area of the hard drive.*

File and Folder Permissions

On NTFS volumes you can set permissions on files and folders that specify which groups and users have access, and what level of access is permitted. NTFS file and folder permissions apply to users on the local computer and to users accessing the file or folder over the network. File and folder permissions are maintained in discretionary access control lists.



Local sharing and security

To share this folder with other users of this computer only, drag it to the Shared Documents folder.

To make this folder and its subfolders private so that only you have access, select the following check box.

☑ Make this folder private

Network sharing and security

To share this folder with both network users and other users of this computer, select the first check box below and type a share name.

□ Share this folder on the network

Share name:

□ Allow network users to change my files

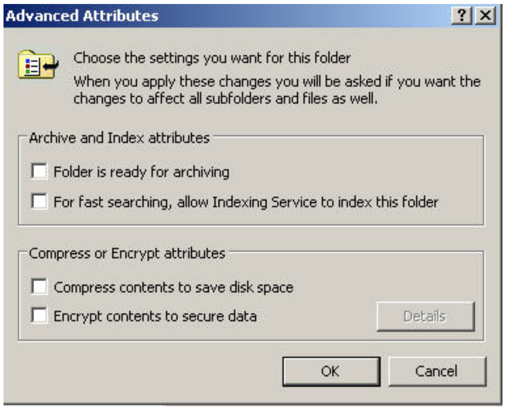
Learn more about sharing and security.

The options on this tab are disabled because this folder is in another folder that is private.

Encryption

The encrypting file system (EFS) uses symmetric key encryption in conjunction with public key technology to protect files and folders. Encryption ensures that only the authorized users and designated recovery agents of that file or folder can access it. Users of EFS are issued a digital certificate with a public key and a private key pair. EFS uses the key set for the user who is logged on to the local computer where the private key is stored.

Users work with encrypted files and folders just as they do with any other files and folders. Encryption is transparent to any authorized users; the system decrypts the file or folder when the user opens it. When the file is saved, encryption is reapplied. However, intruders who try to access the encrypted files or folders receive an "Access denied" message if they try to open, copy, move, or rename the encrypted file or folder.



Choose the settings you want for this folder

When you apply these changes you will be asked if you want the changes to affect all subfolders and files as well.

Archive and Index attributes

□ Folder is ready for archiving

□ For fast searching, allow Indexing Service to index this folder

Compress or Encrypt attributes

□ Compress contents to save disk space

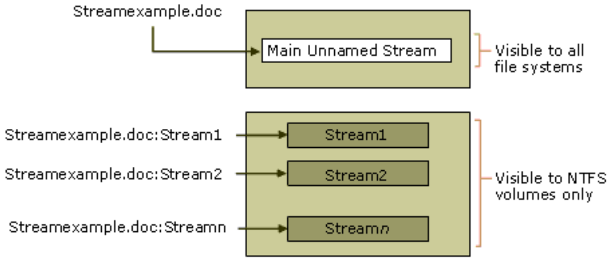
□ Encrypt contents to secure data

Larger Volume Size

The maximum NTFS volume size as implemented in Windows XP Professional is 232 clusters minus 1 cluster, which is approximately 256 terabytes with a max individual file size of about 16 terabytes. Under FAT32, the maximum volume size was 32Gig with a 4 Gig file. This has considerable impact on storage requirement for making forensic duplications and putting together fregmented files.

Multiple Data Streams

A data stream is a sequence of bytes. An application populates the stream by writing data at specific offsets within the stream. The application can then read the data by reading the same offsets in the read path. Every file has a main, unnamed stream associated with it, regardless of the file system used. However, NTFS supports additional named data streams in which each data stream is an alternate sequence of bytes as illustrated in the figure. Applications can create additional named streams and access the streams by referring to their names. This feature permits related data to be managed as a single unit. For example, a graphics program can store a thumbnail image of bitmap in a named data stream within the NTFS file containing the image.


Streamexample.doc
Main Unnamed Stream
Visible to all file systems
Streamexample.doc:Stream1
Stream1
Streamexample.doc:Stream2
Stream2
Streamexample.doc:Streamn
Streamn
Visible to NTFS volumes only

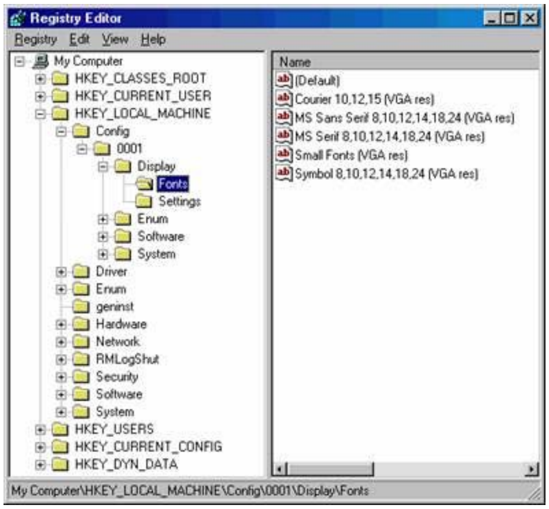
A forensic examiner is particularly interested in these multiple data streams since they can hide data either intentionally or by coincidence. The data stream is an additional data attribute of a file.

Cluster Size

As described previously, the cluster size has also significantly increased with NTFS.

CERT Training and Education

Windows Registry



My Computer\HKEY_LOCAL_MACHINE\Config\0001\Display\Fonts

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.7 Windows Registry

The registry [Russinovich 97] has been the centralized configuration database for Windows and supported applications since Windows 95s' introduction. From the Windows registry, a forensics examiner can discover hardware and software configuration, user preferences, initial system settings, and even logon and password information. Windows artifact analysis is quickly becoming an important part of forensic discovery, so you must understand how altering the registry settings can corrupt the system's evidence collection and documentation.

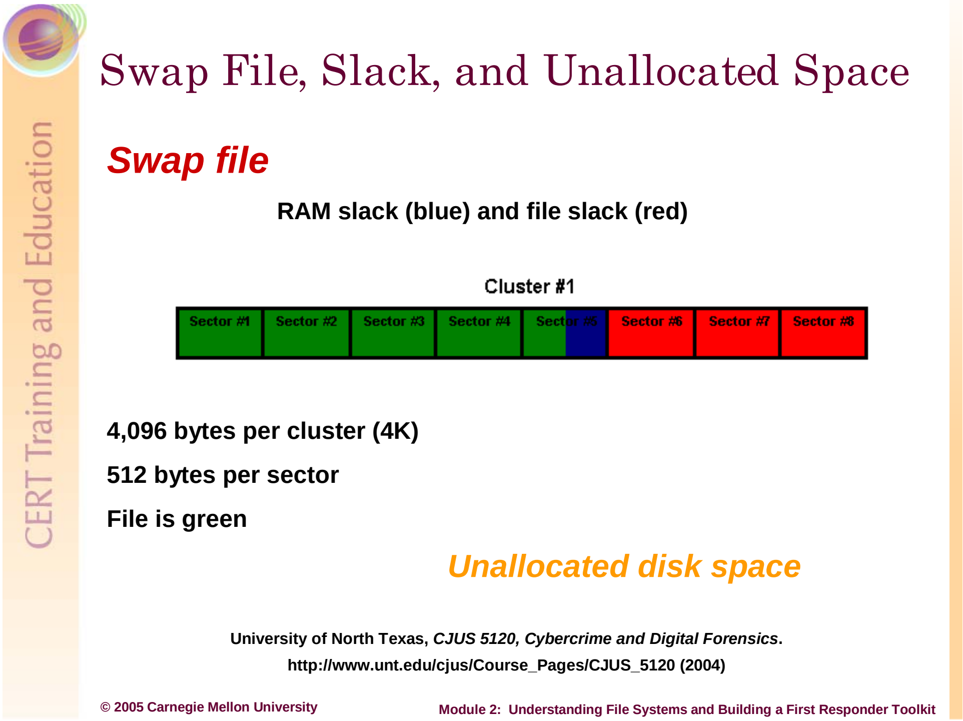
Over time, as more and more installation and removal program settings are saved, the registry becomes cluttered. The following freeware utilities can delete these unused or old registry keys and clean up your computer:

- **Jv16 Power Tools v1.4.1.248**

  http://www.pcworld.com/downloads/file_description/0,fid,22765,00.asp

- **RegClean v4.1a**

  http://www.pcworld.com/downloads/file_description/0,fid,4666,RSS,RSS,00.asp



CERT Training and Education

Swap File, Slack, and Unallocated Space

Swap file

RAM slack (blue) and file slack (red)

4,096 bytes per cluster (4K)

512 bytes per sector

File is green

Unallocated disk space

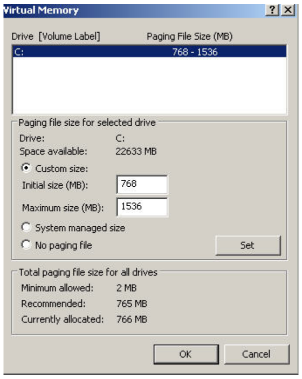
University of North Texas, *CJUS 5120, Cybercrime and Digital Forensics*.
http://www.unt.edu/cjus/Course_Pages/CJUS_5120 (2004)

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.8 Swap File, Slack, and Unallocated Space

2.2.8.1 Swap File

The swap file is a hidden system file that is used for virtual memory when there is not enough physical memory to run programs. Space on the hard drive is temporarily swapped with the RAM as programs are running. This swap file contains portions of all documents and other material a user produces while using the computer. In Windows NT/XP systems, the swap file is named pagefile.sys; on older Windows 9X systems, it is named win386.swp. The key for a forensic examiner is that a user might cover his tracks by wiping all files but this one, which he may not know exists.



Drive [Volume Label]	Paging File Size (MB)
C:	768 - 1536

Paging file size for selected drive

Drive: C:
Space available: 22633 MB
● Custom size:
Initial size (MB): 768
Maximum size (MB): 1536
○ System managed size
○ No paging file

Total paging file size for all drives

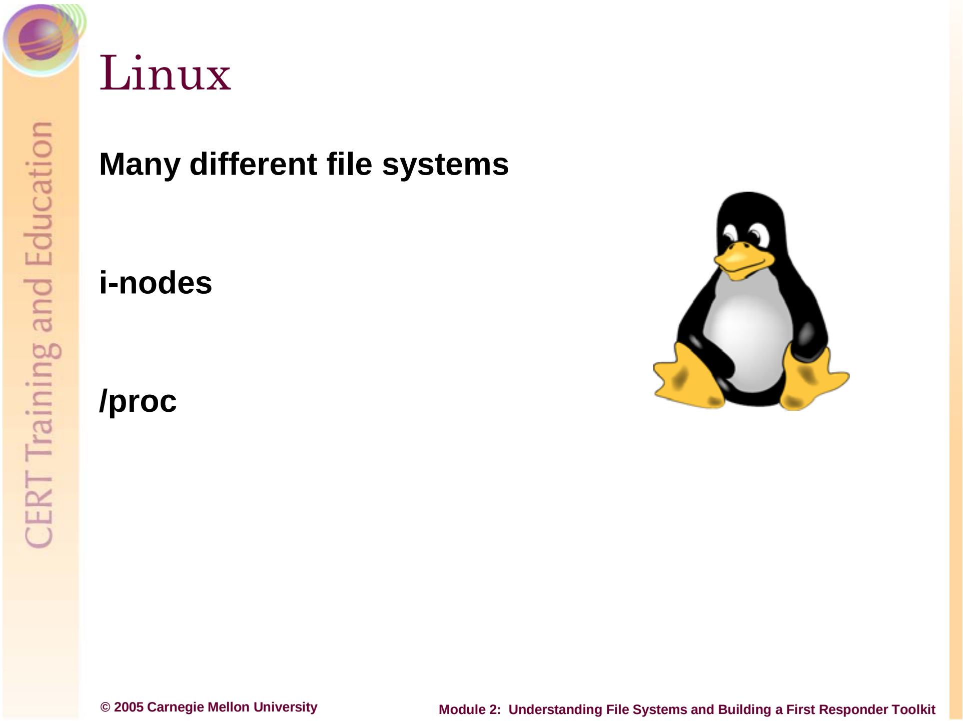
Minimum allowed: 2 MB
Recommended: 765 MB
Currently allocated: 766 MB

2.2.8.2 Slack Space

As shown in the slide's graphic, slack space results when file systems create a cluster (Windows) or block (Linux) but do not necessarily use the entire fixed length space that was allocated. In this case, file information from previous use is still available, long after deletions and rewrites. At the very least, viewing these clusters can prove to a forensic examiner that a certain type of file might have existed. A significant advantage of NTFS is that it generates much less slack space.

2.2.8.3 Unallocated Space

When a user deletes a file, it is flagged as no longer needed, but it remains on the system until it is overwritten. The remaining files are in unallocated disk space, where clusters/blocks are not assigned but may contain data. A forensic software tool can identify and restore these files.



CERT Training and Education

Linux

- **Many different file systems**
- **i-nodes**
- **/proc**

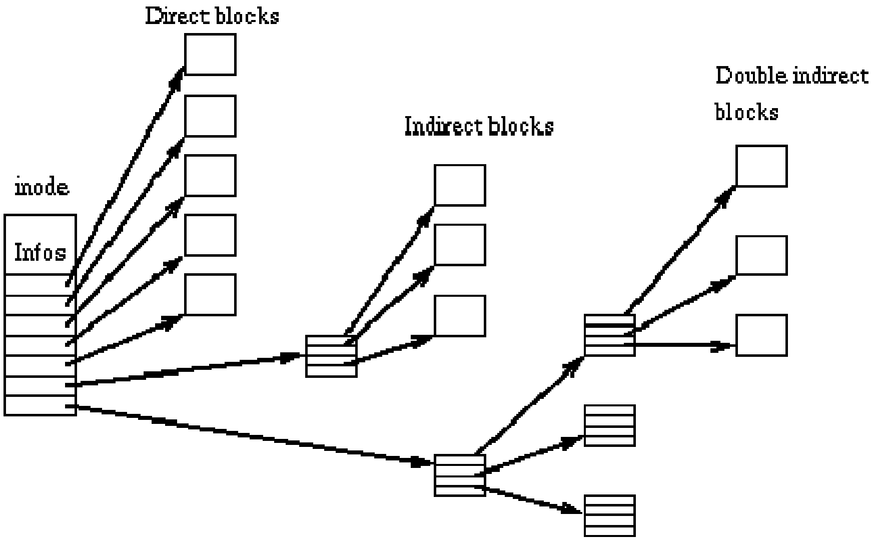
© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.9 Linux File System Basics

The following excerpt is from a paper describing the Linux basic file system first published in the *Proceedings of the First Dutch International Symposium on Linux* (http://e2fsprogs.sourceforge.net/ext2intro.html):

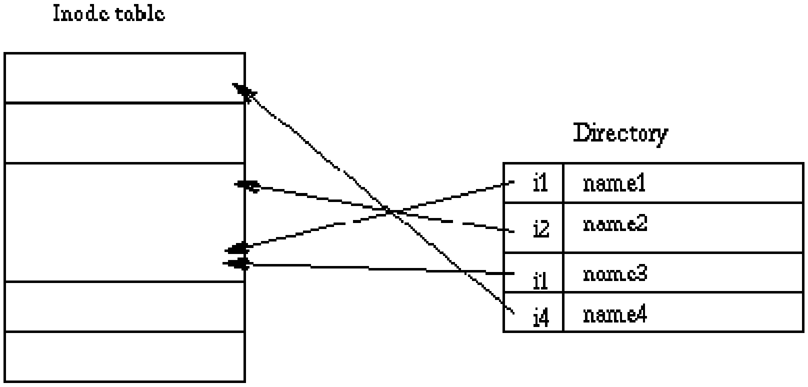
> ***Basic File System Concepts***
>
> *Every Linux file system implements a basic set of common concepts derivated from the UNIX OS. Files are represented by inodes, and directories are simply files containing a list of entries and devices can be accessed by requesting I/O on special files.*
>
> *Inodes*
>
> *Each file is represented by a structure, called an inode. Each inode contains the description of the file: file type, access rights, owners, timestamps, size, pointers to data blocks. The addresses of data blocks allocated to a file are stored in its inode. When a user requests an I/O operation on the file, the kernel code converts the current offset to a block number, uses this number as an index in the block addresses table and reads or writes the physical block. This figure represents the structure of an inode:*



Directories

Directories are structured in a hierarchical tree. Each directory can contain files and subdirectories.

Directories are implemented as a special type of files. Actually, a directory is a file containing a list of entries. Each entry contains an inode number and a file name. When a process uses a pathname, the kernel code searches in the directories to find the corresponding inode number. After the name has been converted to an inode number, the inode is loaded into memory and is used by subsequent requests.



Links

UNIX file systems implement the concept of link. Several names can be associated with an inode. The inode contains a field containing the number associated with the file. Adding a link simply consists of creating a directory entry, where the inode number points to the inode, and in incrementing the links *count in the inode. When a link is deleted, i.e. when one uses the* **`rm`** *command to remove a filename, the kernel decrements the links count and deallocates the inode if this count becomes zero.*

This type of link is called a hard link and can only be used within a single file system: it is impossible to create cross-file system hard links. Moreover, hard links can only point on files: a directory hard link cannot be created to prevent the apparition of a cycle in the directory tree.

Another kind of link exists in most UNIX file systems. Symbolic links are simply files which contain a filename. When the kernel encounters a symbolic link during a pathname to inode conversion, it replaces the name of the link by its contents, i.e. the name of the target file, and restarts the pathname interpretation. Since a symbolic link does not point to an inode, it is possible to create cross-file systems symbolic links. Symbolic links can point to any type of file, even on nonexistent files. Symbolic links are very useful because they don't have the limitations associated to hard links. However, they use some disk space, allocated for their inode and their data blocks, and cause an overhead in the pathname to inode conversion because the kernel has to restart the name interpretation when it encounters a symbolic link.

Device Special Files

In UNIX-like OSs, devices can be accessed via special files. A device special file does not use any space on the file system. It is only an access point to the device driver.

Two types of special files exist: character and block special files. The former allows I/O operations in character mode while the later requires data to be written in block mode via the buffer cache functions. When an I/O request is made on a special file, it is forwarded to a (pseudo) device driver. A special file is referenced by a major number, which identifies the device type, and a minor number, which identifies the unit.

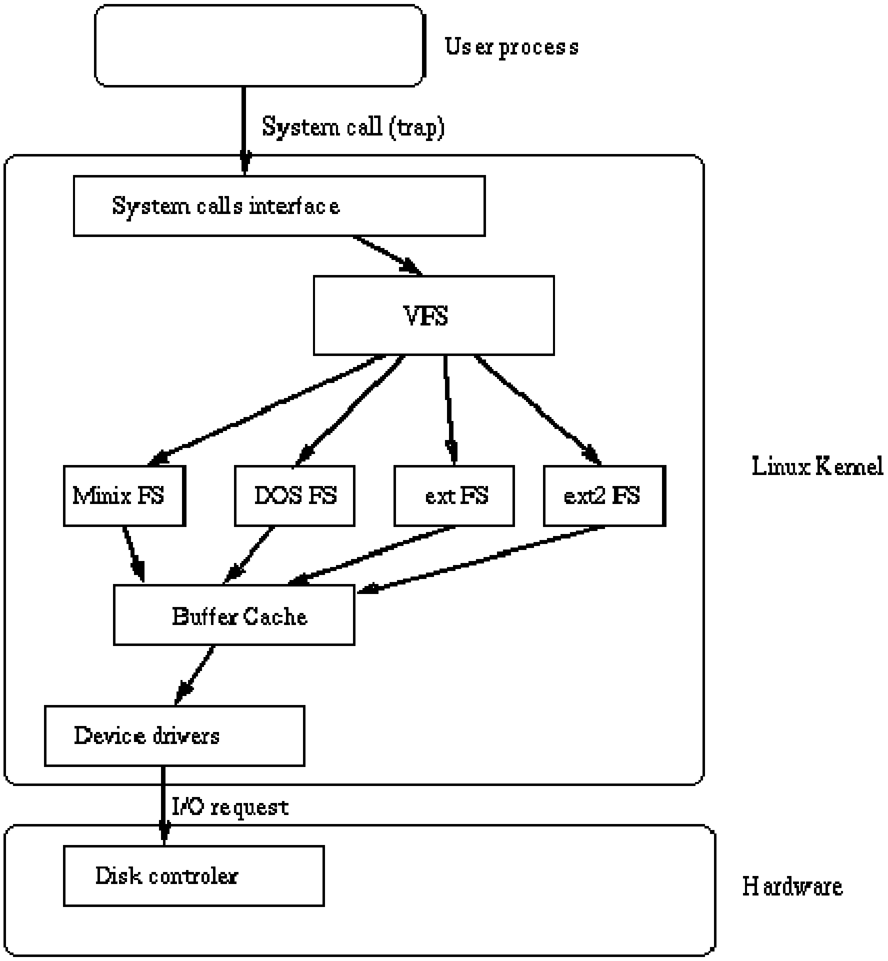
The Virtual File System

Principle

The Linux kernel contains a virtual file system layer which is used during system calls acting on files. The VFS is an indirection layer which handles the file oriented system calls and calls the necessary functions in the physical file system code to do the I/O.

This indirection mechanism is frequently used in UNIX-like OSs to ease the integration and the use of several file system types.

When a process issues a file oriented system call, the kernel calls a function contained in the VFS. This function handles the structure independent manipulations and redirects the call to a function contained in the physical file system code, which is responsible for handling the structure dependent operations. File system code uses the buffer cache functions to request I/O on devices. This scheme is illustrated in this figure:



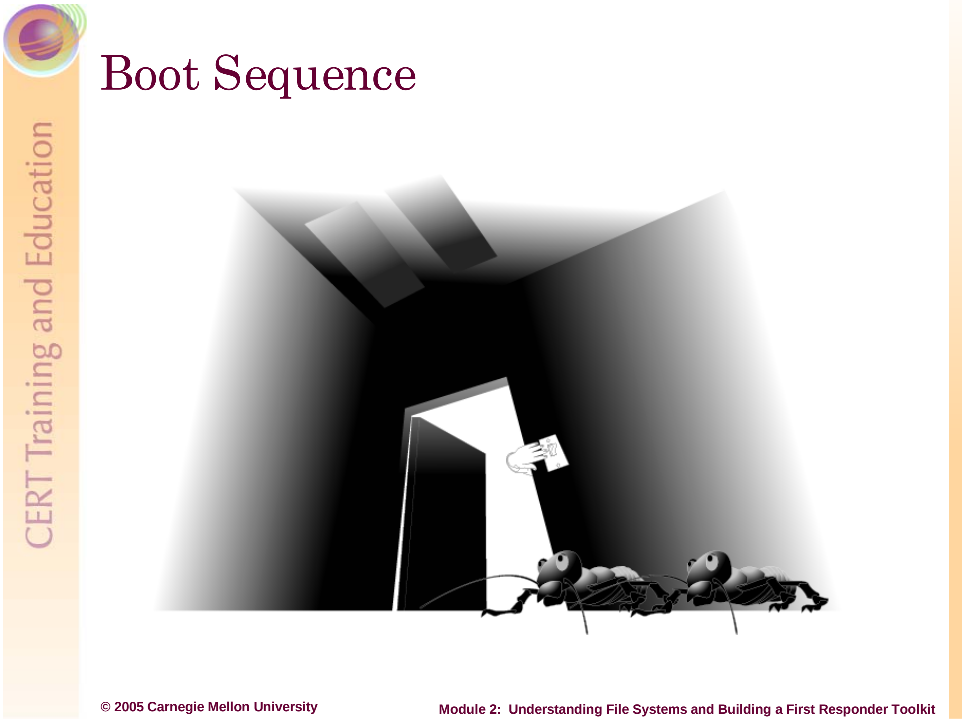
The VFS Structure

The VFS defines a set of functions that every file system has to implement. This interface is made up of a set of operations associated to three kinds of objects: file systems, inodes, and open files.

The VFS knows about file system types supported in the kernel. It uses a table defined during the kernel configuration. Each entry in this table describes a file system type: it contains the name of the file system type and a pointer on a function called during the mount operation. When a file system is to be mounted, the appropriate mount function is called. This function is responsible for reading the superblock from the disk, initializing its internal variables, and returning a *mounted file system descriptor to the VFS. After the file system is mounted, the VFS functions can use this descriptor to access the physical file system routines.*

A mounted file system descriptor contains several kinds of data: information that is common to every file system type, pointers to functions provided by the physical file system kernel code, and private data maintained by the physical file system code. The function pointers contained in the file system descriptors allow the VFS to access the file system internal routines.

Two other types of descriptors are used by the VFS: an inode descriptor and an open file descriptor. Each descriptor contains information related to files in use and a set of operations provided by the physical file system code. While the inode descriptor contains pointers to functions that can be used to act on any file (e.g., `create`*,* `unlink`*), the file descriptors contains pointer to functions which can only act on open files (e.g.,* `read`*,* `write`*).*



CERT Training and Education

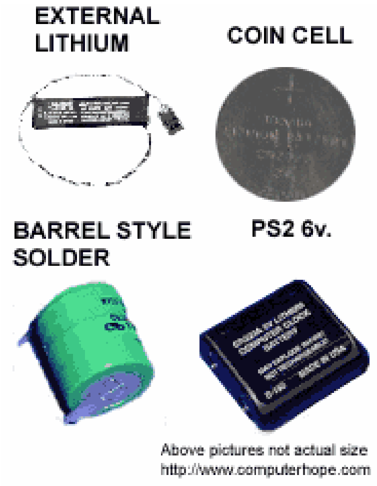
Boot Sequence

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.10 Boot Sequence

To ensure that forensic data is not contaminated or altered, a first responder needs to understand how to open and set the suspicious computer's CMOS (complementary metal-oxide semiconductor) and BIOS (basic input/output system) to boot from a CDROM. Note the BIOS lives inside the CMOS Chip. The CMOS uses a small battery to store system configuration information, including the date and time, when the power is off. The BIOS contains programs that include the instructions for input and output at the hardware level. If a system has a BIOS password usually this can be reset by removing the CMOS battery.



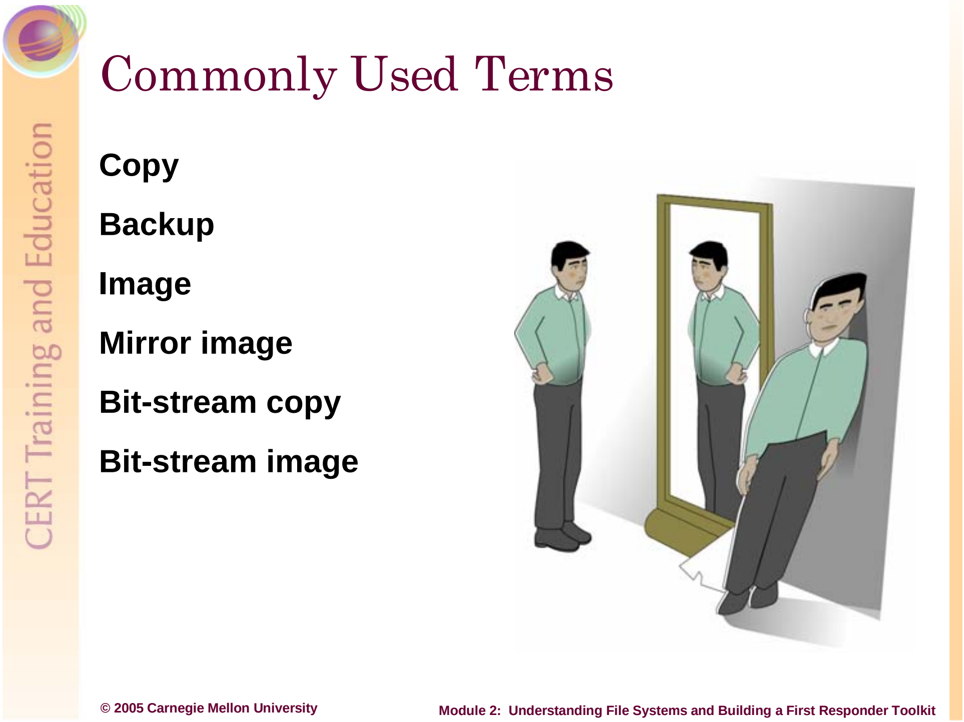
Above pictures not actual size
http://www.computerhope.com

Figure 3: Types of CMOS Batteries

When a system is restarted, the boot sequence and other start-up activities can modify important forensic data. Therefore, you must enable the BIOS to boot directly from a trusted floppy or CD drive. On most systems, during start-up, the screen usually displays the appropriate key (i.e., Del, Ctrl-Alt-Ins, or a function key) that brings up the BIOS menu. Read the suspicious computer’s instruction manual so that you access the BIOS set-up on start-up and prevent unnecessary frustration and possible modification of data. This is an important tip in your pre-incident response preparation: know your systems and maintain a library of manuals and documentation. Many vendors also have CMOS and BIOS information and instruction on their Web site.

For more details, visit the following Web sites:

- http://www.pcguide.com/ref/mbsys/bios/boot.htm
- http://computer.howstuffworks.com/bios.htm
- http://www.computerhope.com/help/cmos.htm



CERT Training and Education

Commonly Used Terms

- Copy
- Backup
- Image
- Mirror image
- Bit-stream copy
- Bit-stream image

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.11 Commonly Used Terms

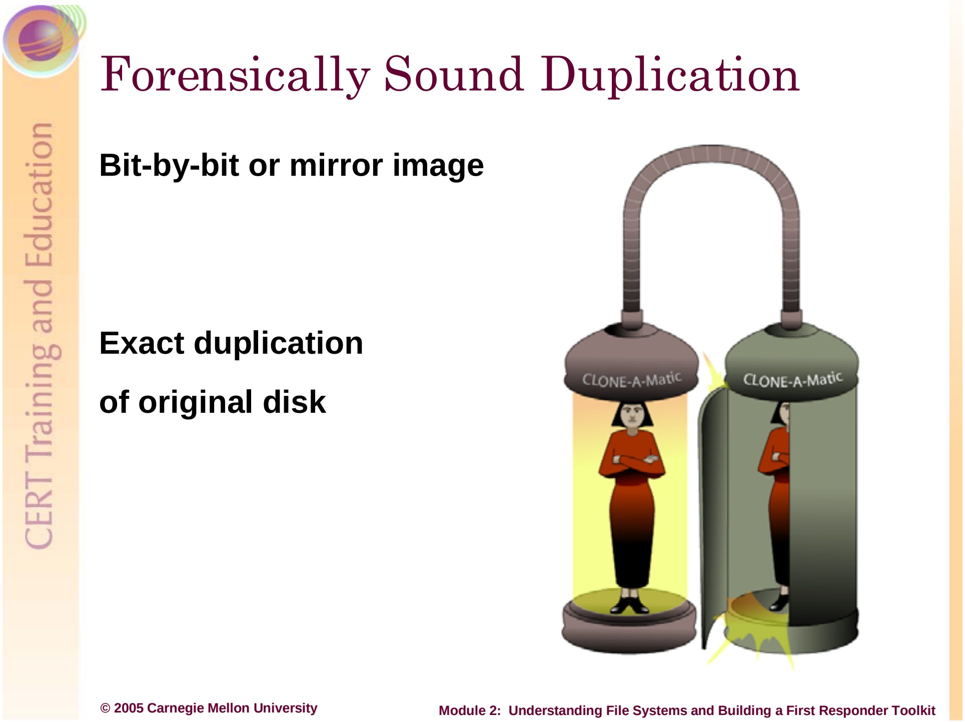
When working with a forensic examiner, clarify the terms “copy” and “image.” For example, if another system administrator says “Copy the drive to make an image,” he probably needs a backup (just files). But if a forensic examiner asks the same question, he probably needs a duplicate image (every 1 and 0 on the disk) for use as forensic evidence. The following definitions are from New Technologies, Inc [NTI 03]:

Copy: Includes only file information, not slack space or unallocated space

Backup: Files copied for future restoration

Image: A file copy of a complete disk used for duplication or restoration

Duplicate image, bit-stream copy, bit-stream image: Exact replica of all sectors, including every 1 and 0. This includes slack space, unallocated space, and the swap file.



CERT Training and Education

Forensically Sound Duplication

Bit-by-bit or mirror image

Exact duplication

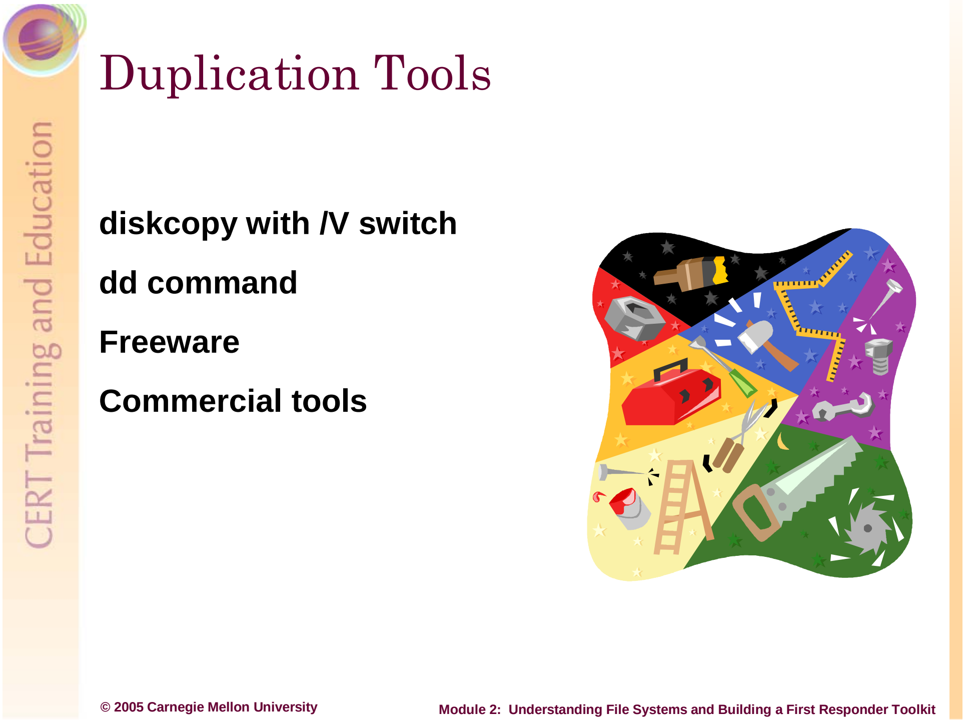
of original disk

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.12 Forensically Sound Duplication

Bit-by-bit copy or bit-stream “duplicate” image means capturing every 1 and 0 on the hard drive, including slack space, unallocated space, and the swap file. Understanding bit-by-bit duplication and why it is forensically sound helps you to decide whether to make a copy (captures active files with the OS’s copy command) or an image (captures active and latent data) of a drive.



CERT Training and Education

Duplication Tools

- diskcopy with /V switch
- dd command
- Freeware
- Commercial tools

© 2005 Carnegie Mellon University

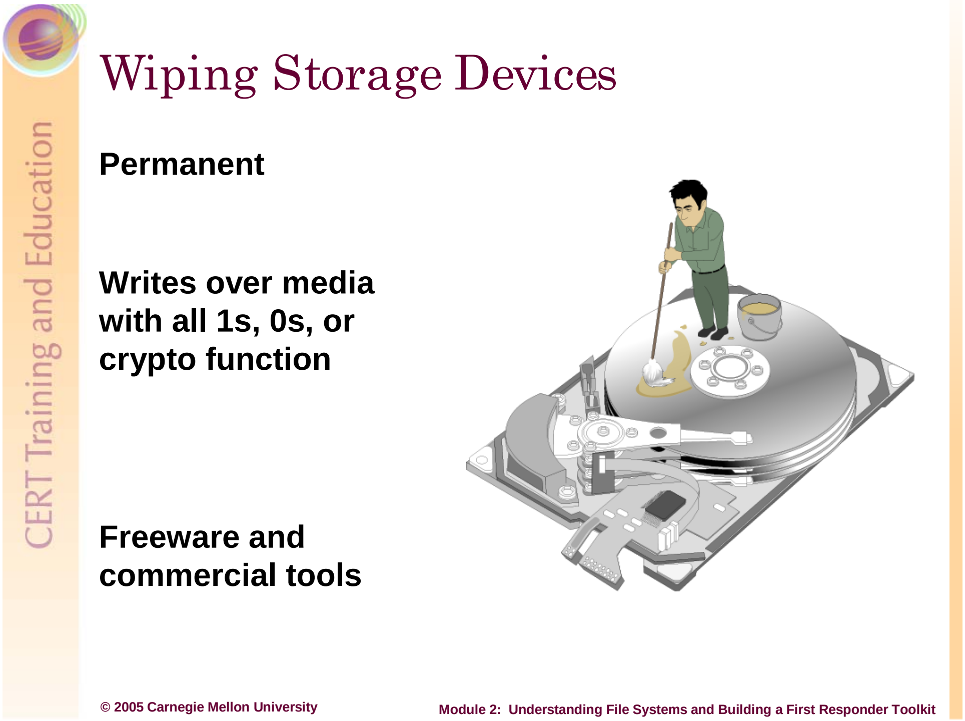
Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.13 Duplication Tools

For a comprehensive list of freeware security and file management tools, visit the following Web sites:

- http://www.cert.org/tech_tips/security_tools.html
- http://www.pcworld.com/downloads/browse/0,cat,1503,sortIdx,1,pg,1,00.asp

More on duplication tools will occur in Module 4.



CERT Training and Education

Wiping Storage Devices

Permanent

Writes over media with all 1s, 0s, or crypto function

Freeware and commercial tools

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

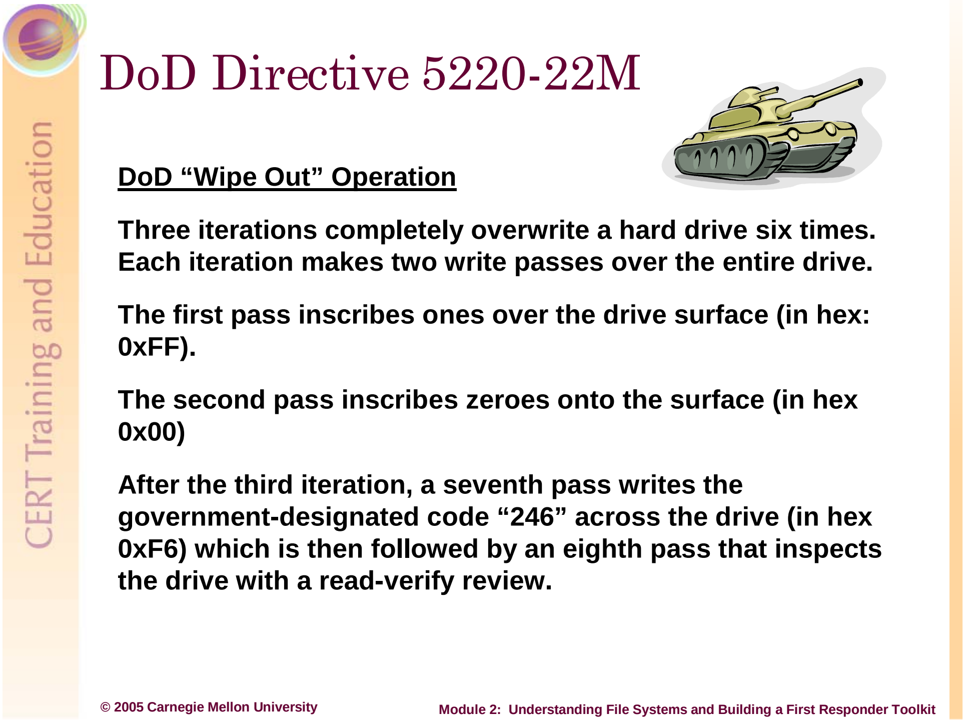
2.2.14 Wiping Storage Devices

An interesting article [MIT 03] demonstrates why disks should be wiped before they are sold, donated, reused, or trashed. Two MIT graduate students looked at 158 used disks and found an assortment of personal and financial information, including over 5,000 credit card numbers.

Recalling from the first part of this module, data that is erased from a hard drive is not necessarily permanently removed. There is a good chance that latent sections remain. The only sure way to be reasonably certain is to wipe the storage device.

For assorted freeware and commercial wipe file/disk products, visit the following Web sites:

- **Active@ Kill Disk Hard Drive Eraser v1.1**
  http://www.pcworld.com/downloads/file_description/0,fid,22920,00.asp
- **Sure Delete v5.1**
  http://www.pcworld.com/downloads/file_description/0,fid,22393,00.asp
- **WipeDrive v3.0.2**
  http://www.pcworld.com/downloads/file_description/0,fid,22961,00.asp
- **Eraser v5.7**
  http://www.pcworld.com/downloads/file_description/0,fid,22963,00.asp
- **PGP**
  http://web.mit.edu/network/pgp.html



CERT Training and Education

DoD Directive 5220-22M

<u>DoD "Wipe Out" Operation</u>

Three iterations completely overwrite a hard drive six times. Each iteration makes two write passes over the entire drive.

The first pass inscribes ones over the drive surface (in hex: 0xFF).

The second pass inscribes zeroes onto the surface (in hex 0x00)

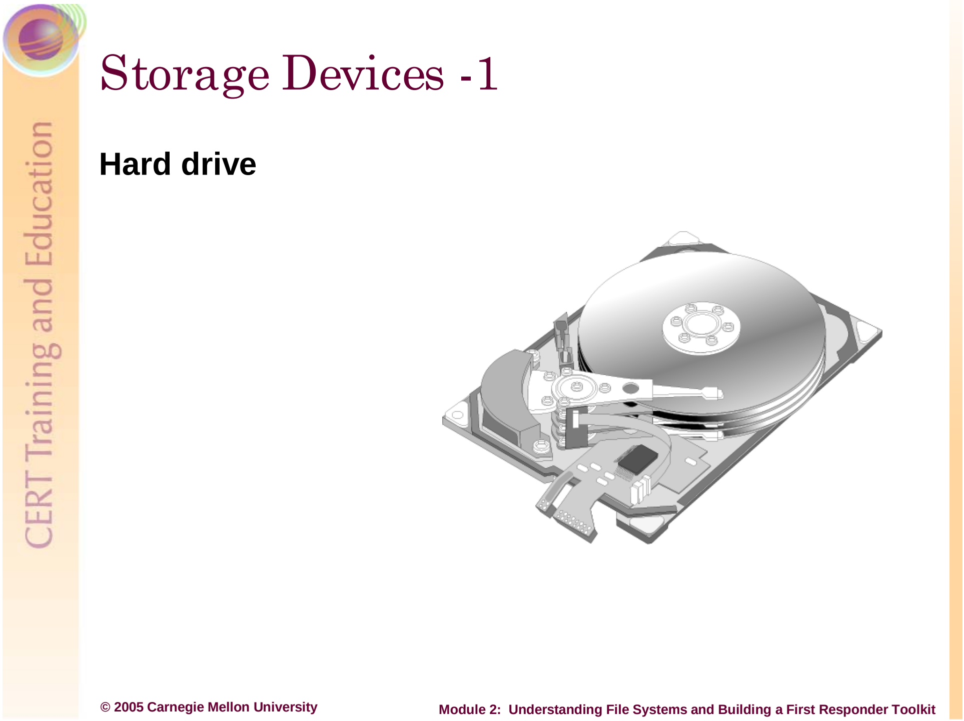
After the third iteration, a seventh pass writes the government-designated code "246" across the drive (in hex 0xF6) which is then followed by an eighth pass that inspects the drive with a read-verify review.

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.15 DoD Directive 5220-22M

The United States Department of Defense (DoD) issued Directive 5220-22M as part of its National Industrial Security Program. The directive sets security standards for DoD contractors for sanitizing hard drives that contain less than top-secret data.

For more details, visit http://www.dss.mil/isec/nispom.htm.



CERT Training and Education

Storage Devices -1

Hard drive

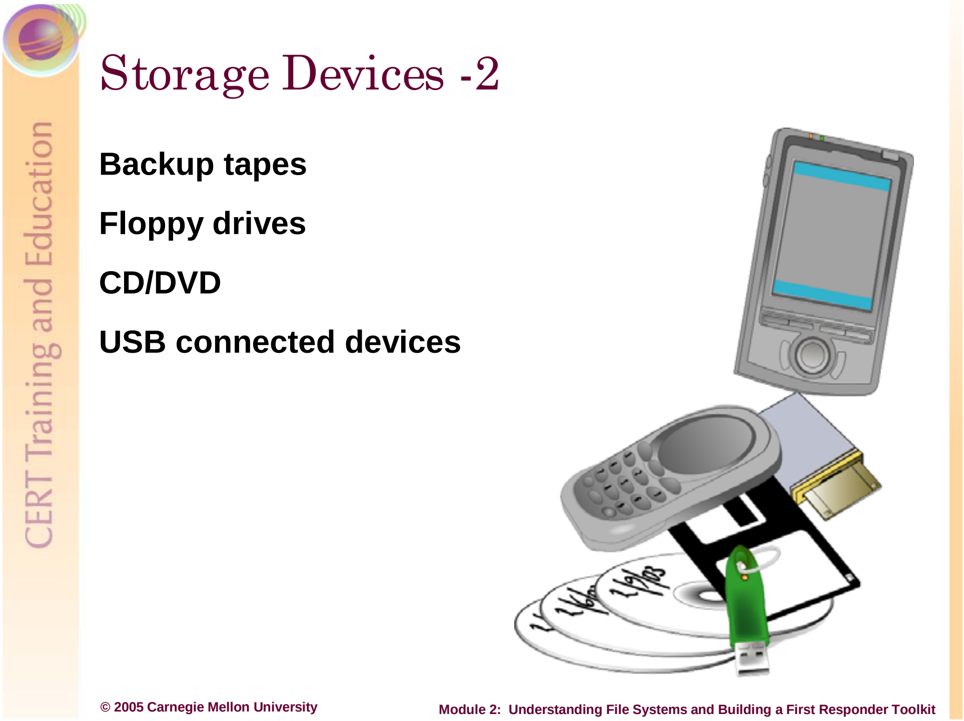
© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.16 Hard Drives

When evidence is expected to be found on a hard drive, these factors need to be addressed:

- Is the drive on the server or a user’s computer? The necessary evidence may only be on one partition, not an entire physical drive. Also a user’s computer might simply be seized as evidence.
- Is the evidence mission critical or can it be removed? Sometimes it is easier to hand over the original hard drive to the investigators, rather then perform the time consuming duplication process. If the evidence is on a mission critical system it may not be possible unless a hot swappable back-up is available.
- Do you have additional sanitized (wiped) hard drives with the adequate storage capacity? It is good to have prepared hard drives on hand of the proper capacity in case duplication must occur before an investigation team arrives.
- Does the evidence span several physical hard drives? (Raid Array, NTFS Extended Volume)



CERT Training and Education

Storage Devices -2

- Backup tapes
- Floppy drives
- CD/DVD
- USB connected devices

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.2.17 Other Storage Devices

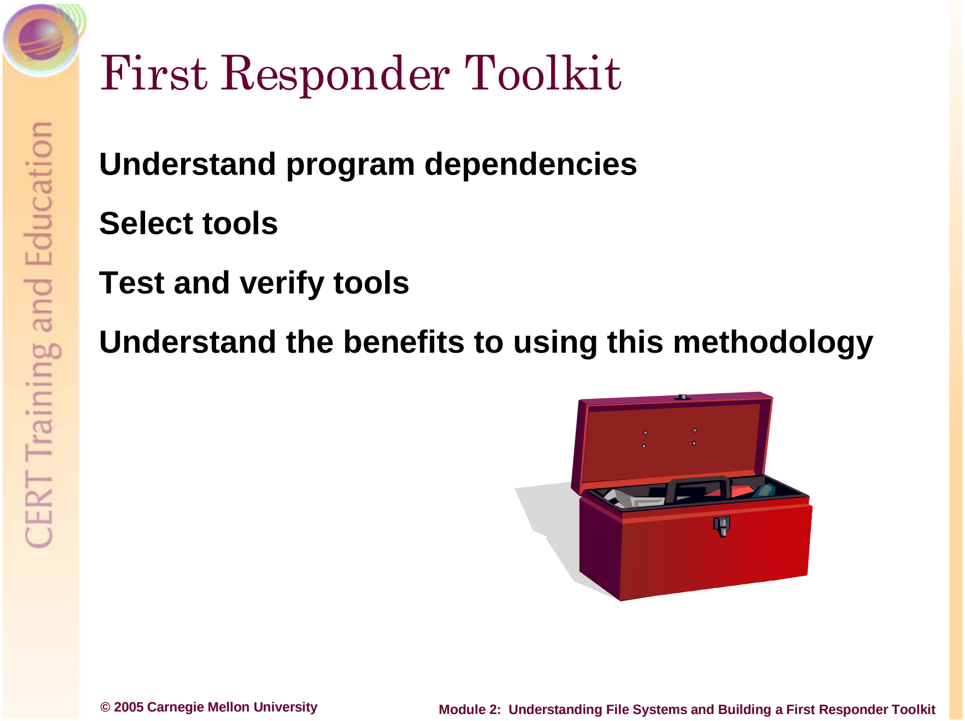
There are many types of storage devices, many of which still use the FAT file system. As shown below, storage devices do not have to have the same file system as the OS it interacts with.

Backup tape: host OS

Floppy drive: FAT12

CD/DVD: ISO 9660

Compact Flash, USB thumb drive: usually FAT32



CERT Training and Education

First Responder Toolkit

Understand program dependencies

Select tools

Test and verify tools

Understand the benefits to using this methodology

© 2005 Carnegie Mellon University

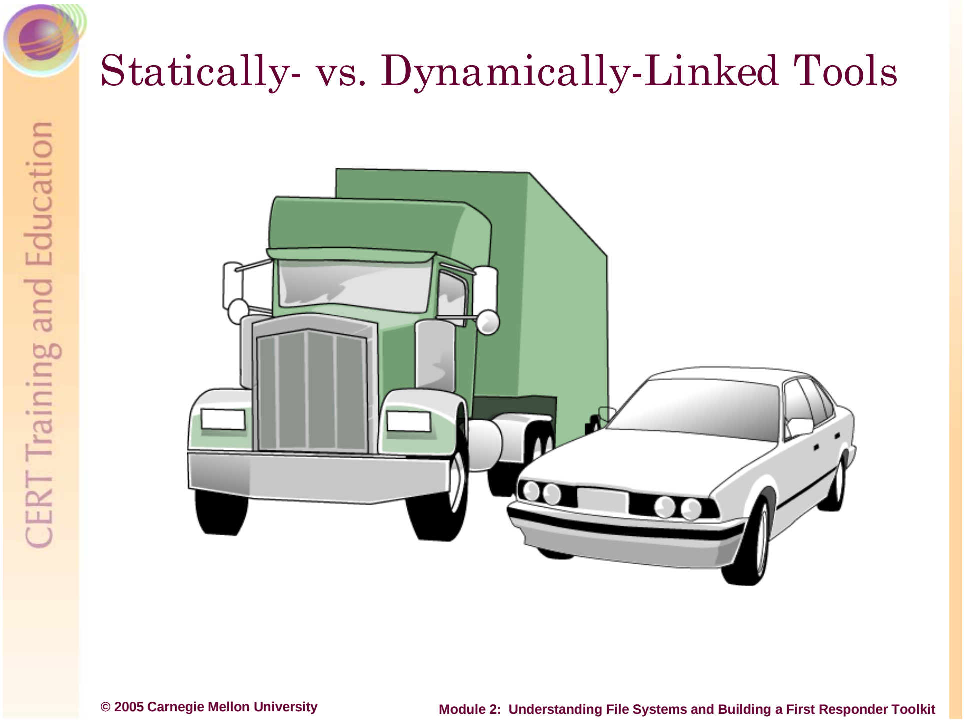
Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3 First Responder Toolkit

It is important to understand that network and security professionals should have trusted tool kits prepared. When responding to a system whose trust state cannot be known, it should be considered untrustworthy until proven otherwise.

When a program is executed, it normally uses shared libraries for routine system commands and changes those common files' access times. This is important for forensic examiners as they examine machines to determine when activities took place. It is this issue that prompts the creation of a trusted tool. A trusted tool should have the needed libraries statically compiled when possible.

Also, as a first responder, you'll want to select tools that create output-specific information and then test it to determine system dependencies so that it can be statically compiled (best case) and well documented.



CERT Training and Education

Statically- vs. Dynamically-Linked Tools

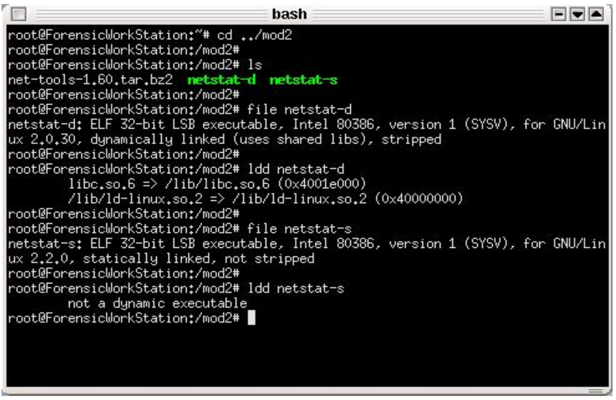
© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3.1 Statically- vs. Dynamically-Linked Tools

The significant difference between a statically-linked executable and a dynamically-linked executable is that a statically-linked executable is self-contained: the program contains all the code necessary to successfully run as a standalone program. Statically-linked executables or tools are forensically better because they limit the impact (footprint) on the suspicious computer.

However, a dynamically-linked executable uses loaded libraries within the computer's memory to run. An example of a dynamically-linked executable that could be used by a forensic examiner is the Windows native *netstat.exe* command. This executable uses multiple dynamic-link libraries (DLLs) in order to properly execute.



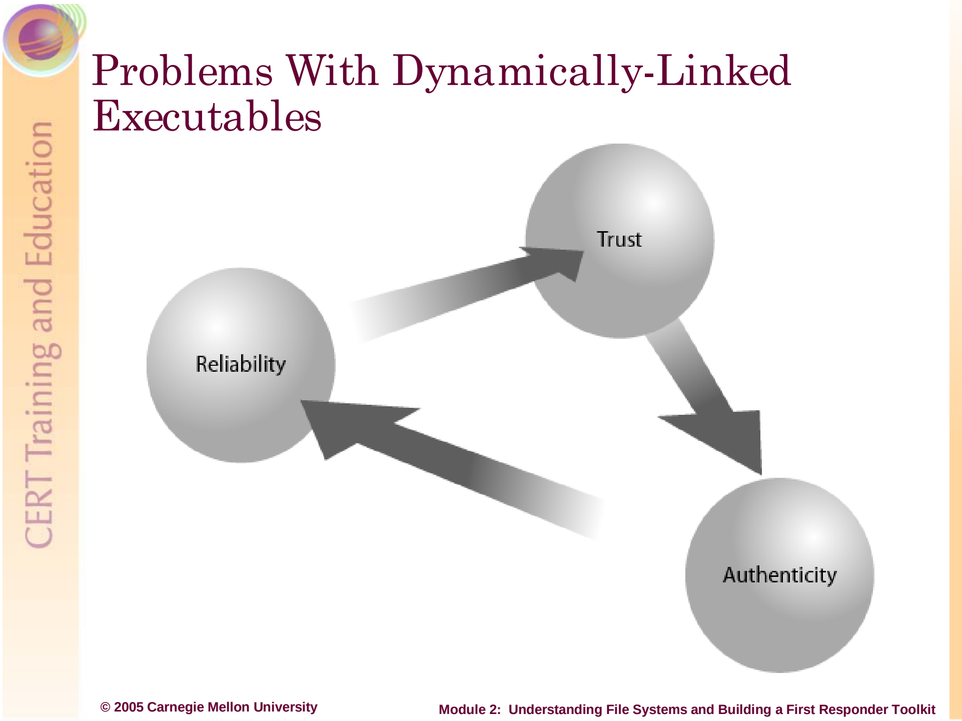
```
root@ForensicWorkStation:~# cd ../mod2
root@ForensicWorkStation:/mod2#
root@ForensicWorkStation:/mod2# ls
net-tools-1.60.tar.bz2  netstat-d  netstat-s
root@ForensicWorkStation:/mod2#
root@ForensicWorkStation:/mod2# file netstat-d
netstat-d: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Lin
ux 2.0.30, dynamically linked (uses shared libs), stripped
root@ForensicWorkStation:/mod2#
root@ForensicWorkStation:/mod2# ldd netstat-d
        libc.so.6 => /lib/libc.so.6 (0x4001e000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
root@ForensicWorkStation:/mod2#
root@ForensicWorkStation:/mod2# file netstat-s
netstat-s: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Lin
ux 2.2.0, statically linked, not stripped
root@ForensicWorkStation:/mod2#
root@ForensicWorkStation:/mod2# ldd netstat-s
        not a dynamic executable
root@ForensicWorkStation:/mod2# █
```

Figure 4: The ldd Command

In dynamic linking, a link is established between the executable and the required libraries. Within a binary, a table called an *import directory* or a *variable-length array of imports* contains the name of each required DLLs. At loadtime, the OS's loader loads the executable and the libraries separately. The loader searches the hard disk for the required libraries (generally located in the Windows System32 folder), loads them into an unpredictable location within memory, and updates the executable with the library's location.

An executable then uses this information to call functions and access data stored in the library. This type of dynamic linking is called loadtime linking and is used by most OSs including Windows and Linux. One disadvantage of dynamic linking is that the executables depend on the separately stored libraries in order to function properly. If the library is deleted, moved, renamed, or replaced with an incompatible or malicious version, the executable could malfunction and cause unreliable and false output [Wikipedia 04].

This further explains why including the necessary DLLs for an executable on a separate disk will not ensure that the binary will use the DLLs on the separate disk. It also explains why the first responder should never trust or use the native commands on the suspicious computer for collecting volatile data.



CERT Training and Education

Problems With Dynamically-Linked Executables

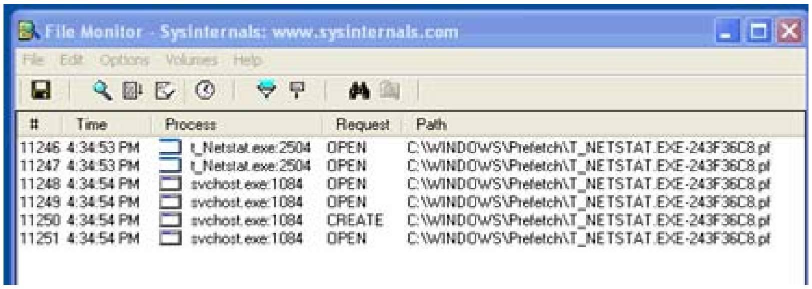
© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3.2 Problems with Dynamically-Linked Executables

One issue in creating a first responder toolkit is the problem with forensic tool dependencies and DLLs in Windows. Because Windows code is closed and not publicly available it is not possible to statically compile Windows native executables. Therefore, at a minimum it is important to define what DLLs and files are affected when executing a Windows program. With this information, it is possible to at least note what impact (or footprint) the program will have on the system.

To determine which files or shared libraries each forensic tool accesses, use the *Filemon* utility. Once the dependencies have been identified, include the list of files on the response disk so that in the event that it becomes necessary, this list can be provided and be used to identify what changes to the system were made by the first responder.

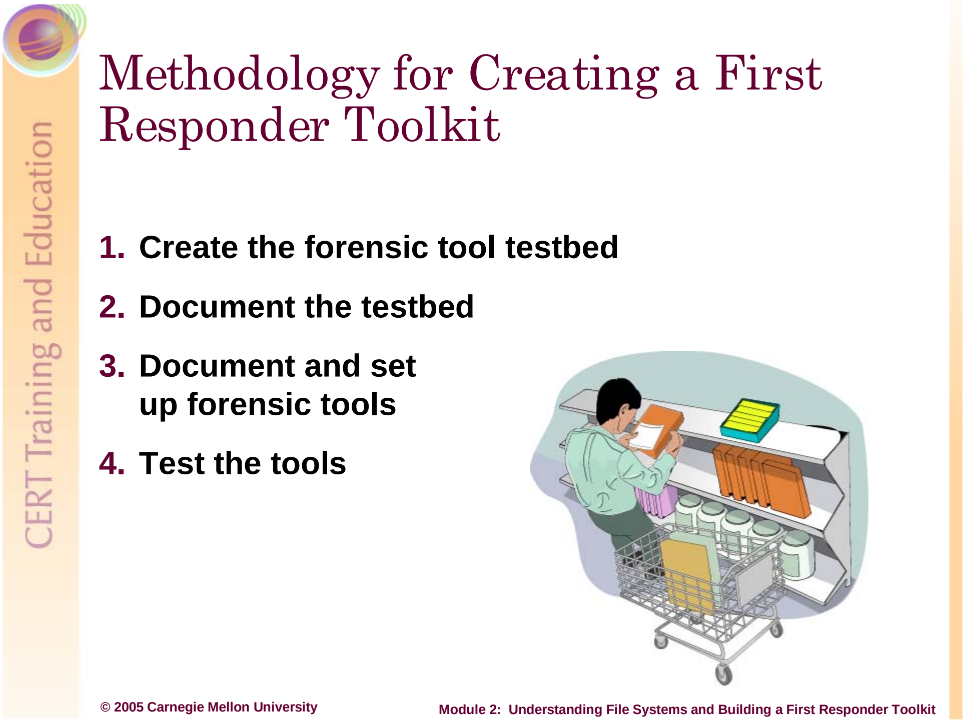


File Monitor - Sysinternals: www.sysinternals.com
File Edit Options Volumes Help

```
#     Time        Process              Request  Path
11246 4:34:53 PM  t_Netstat.exe:2504   OPEN     C:\WINDOWS\Prefetch\T_NETSTAT.EXE-243F36C8.pf
11247 4:34:53 PM  t_Netstat.exe:2504   OPEN     C:\WINDOWS\Prefetch\T_NETSTAT.EXE-243F36C8.pf
11248 4:34:54 PM  svchost.exe:1084     OPEN     C:\WINDOWS\Prefetch\T_NETSTAT.EXE-243F36C8.pf
11249 4:34:54 PM  svchost.exe:1084     OPEN     C:\WINDOWS\Prefetch\T_NETSTAT.EXE-243F36C8.pf
11250 4:34:54 PM  svchost.exe:1084     CREATE   C:\WINDOWS\Prefetch\T_NETSTAT.EXE-243F36C8.pf
11251 4:34:54 PM  svchost.exe:1084     OPEN     C:\WINDOWS\Prefetch\T_NETSTAT.EXE-243F36C8.pf
```

Figure 5: Using Filemon *to Identify Dependencies*

The issue with DLLs is not as pressing in the UNIX world. Since most of the operating systems are open source it is possible to statically build binaries that will not rely on shared libraries. This is the preferred type of tool because its output is more trustworthy.



CERT Training and Education

Methodology for Creating a First Responder Toolkit

1. Create the forensic tool testbed
2. Document the testbed
3. Document and set up forensic tools
4. Test the tools

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3.3 Methodology for a Creating First Responder Toolkit

Before a security incident occurs and before any evidence collection is done, the first responder must develop a toolkit. In most computer crime cases, the first responder will be the system administrator or a member of law enforcement. In either case, being prepared beforehand is crucial; you do not want to waste time putting together a set of tools at the last minute that you *think* might do the job. As you create your toolkit, you will become more familiar with forensic collection tools and native commands and their proposed functionality.

A working knowledge of the tools in your toolkit will aid your collection efforts and help you understand your collection limitations and capabilities.

The methodology for creating a first responder toolkit includes of the following four procedures:

1. create a forensic tool testbed
2. document the testbed
3. document and set up the forensic tools
4. test the tools

As a first responder, you should follow this methodology to ensure the integrity and reliability of each collection tool, command, or application you use in the field. It provides guidelines and reproducible procedures for future toolkit verification.

CERT Training and Education

Create a Forensic Tool Testbed

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3.3.1 Create a Forensic Tool Testbed

The first step in creating a first responder toolkit is to create a trusted forensic computer or testbed. This testbed will be used to test the execution and functionality of each forensic collection tool you consider using. But before any tool is tested, ensure that the forensic tool testbed is a trusted resource. It should be a fresh install, free of artifacts from any previous use that could compromise its integrity and reliability. The testbed should be comparable to the systems that you are responding to.

Identify the OS Type

Choose an OS type. If the enterprise you respond to is primarily Windows based, it makes sense to use a Windows forensic tool testbed. The same holds true for Linux. If the enterprise has Windows and Linux machines, you may want to create two different testbed machines, one for each type of OS you may be dealing with. However, VMware allows you to install multiple OSs on a single machine.

Sanitize the Testbed

Completely sanitize the machine so that you can start with a fresh slate. This includes a complete (forensic) wipe of the hard drive(s) to remove any data or artifacts left from the machine’s previous use. The following freeware is available.

- **BCWipe** (Windows)

  http://www.jetico.com/index.htm#/bcwipe3.htm

- **Wipe** (Linux)

  http://wipe.sourceforge.net/

Install Software

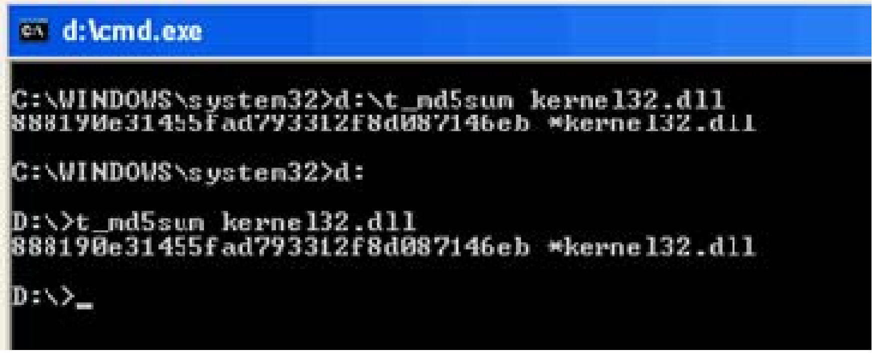
Install the OS and necessary software applications from trusted resources (i.e., designated CD-ROMs for each application and OS). If you have downloaded open source OSs (e.g., Linux variants), then you must verify the hashes before installing the OS. Typically, the testbed system has a minimal set of software applications installed, but it's better for the tool testbed system to replicate a critical system within your IT and network infrastructure so that you know how to respond to machines in that similar state or that have critical services running.

Update and Patch

Bring the system up to date. This is just using best practices and you should treat this testbed machine like any other machine in your IT and network infrastructure. Any necessary patches or hot fixes should be applied to the machine. Again these updates and patches must be applied offline and from a designated disk with known hashes. This 'patch CD' should become a documented item in your tool kit.

Hash the DLLs

For Windows testbed machines, perform a cryptographic hash of the installed DLLs on the system while off-line. This produces checksums for the DLLs at their initial known safe state. After you run and test your forensic tools, you can rehash and compare before and after checksums to determine if any DLLs have been modified or accessed.



```
C:\WINDOWS\system32>d:\t_md5sum kernel32.dll
888190e31455fad793312f8d087146eb *kernel32.dll

C:\WINDOWS\system32>d:

D:\>t_md5sum kernel32.dll
888190e31455fad793312f8d087146eb *kernel32.dll

D:\>_
```

Figure 6: Performing a Cryptographic Hash of Installed DLLs

For Linux testbed machines, you may want to identify all shared libraries on the system and compute a hash of all the shared libraries. Again, rehash the shared libraries after running your tools and then compare checksums.

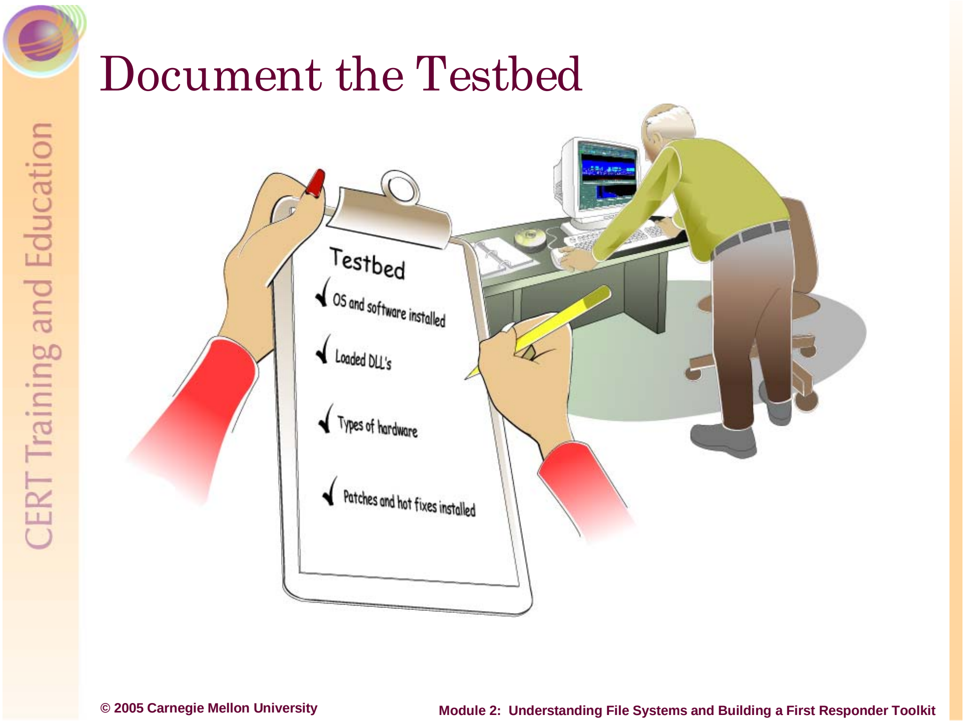
Install a File Integrity Monitor

Using a file integrity monitor on the testbed system is a best practice. File integrity monitors like *Tripwire* or *LANguard System Integrity Monitor (SIM)* should be installed on the forensic testbed to monitor the integrity of the testbed computer's file system. Later, during tool tests, *Tripwire* or *LANguard* can identify changes to MAC times.

Another file system utility called *Filemon* monitors which files each forensic tool uses or references.

For more details, visit the following Web sites:

- http://www.sysinternals.com/ntw2k/utilities.shtml
- http://www.tripwire.org/downloads/index.php
- http://www.gfi.com/languard/



Document the Testbed

CERT Training and Education
Testbed
OS and software installed
Loaded DLL's
Types of hardware
Patches and hot fixes installed

© 2005 Carnegie Mellon University

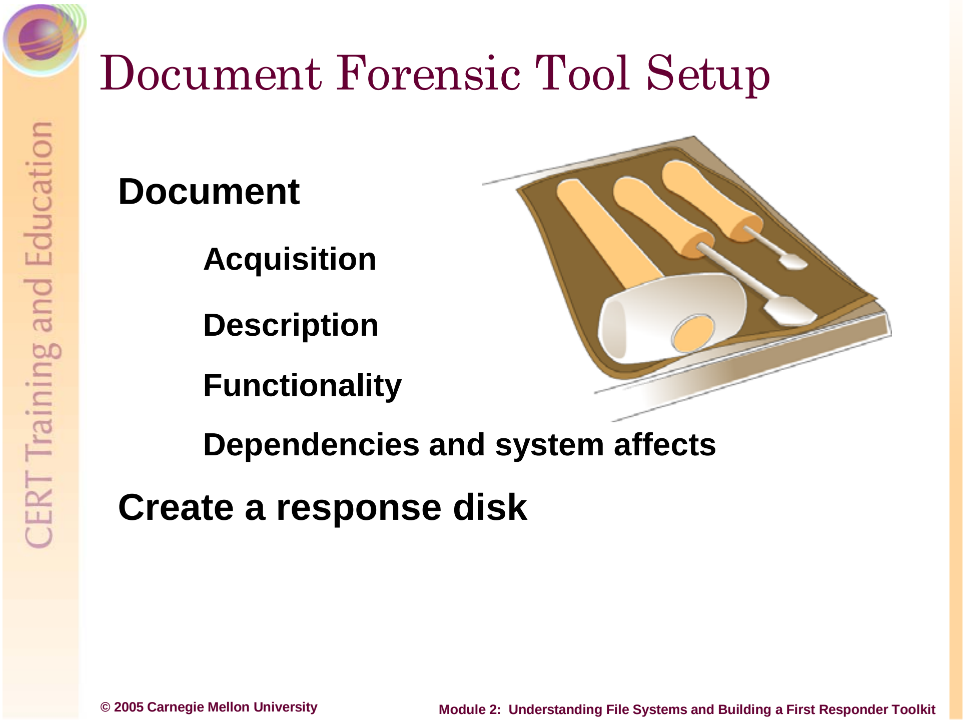
Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3.3.2 Document the Testbed

Documenting the testbed computer(s) is as important as creating them. If you respond to a security incident and the integrity and reliability of your tools and collection procedures are questioned, an independent expert could use your documentation to easily reproduce the situation.

The forensic tool testbed profile includes the following information:

- type of OS
- version of OS (i.e., kernel)
- number and types of applications
- number and types of patches and hot fixes
- types of hardware installed
- MD5 hash of DLLs/shared libraries



CERT Training and Education

Document Forensic Tool Setup

- Document
  - Acquisition
  - Description
  - Functionality
  - Dependencies and system affects
- Create a response disk

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3.3.3 Document and Set Up the Forensic Tools

Creating a profile of forensic tools is as important as creating a profile for the testbed. For each tool that you identify to test, document the following characteristics: acquisition, description, functionality, and dependencies and system affects. By creating a profile of each tool you plan to test, you become familiar with each tool and understand how you can use or tailor the tool to your incident response needs.

Document the Tools

Acquisition: Document how and why you acquired each tool. Identify whether you acquired the tool by downloading, developing, or purchasing it. Document how you verified the integrity of the tool while acquiring it (i.e., verified the MD5 hash for downloaded tools, or testing/validation done for developed/purchased tools).

Description: Document details including—but not limited to—what OS the tool is compatible with, how the tool operates, and how you use the tool (command line vs. GUI).

Functionality: Document as many of the following as apply: the expected generated output; volatile information that can be collected; command-line switches or syntax.

Dependencies and System Affects: Tool dependencies and system affects are probably the two most important tool characteristics to identify and document. You can discover a preliminary set of tool dependencies and system affects by reading the tool's description or read me file, but you will not discover the complete set until you test the tool.

Tool dependencies include required access levels and the use of shared libraries. They are crucial and could limit your forensic collection efforts. Therefore, if you do not have administrative or root credentials, you probably will not want to use tools that require that access.

As you document file dependencies like shared libraries, DLLs, and registry keys, you will recognize what kind of footprint the tool leaves on a system and can use that information as follows.

- For Linux systems, you can statically recompile the command so that it will no longer reference the system.
- For Windows systems, you cannot recompile but you can rank tools based on which has the fewest dependencies and system affects.

Create a Response Disk

Tools should reside on a separate storage media device like a floppy disk, thumb drive, or CD-ROM. (For simplicity, we assume disk and refer to the "response disk.") Which one you choose depends upon space requirements, how many tools you have, and what you think you will have access to when responding to an incident. Systems will generally have a USB port, CD-ROM drive, or floppy drive, but it is a good idea to have your forensic tools on multiple storage devices.

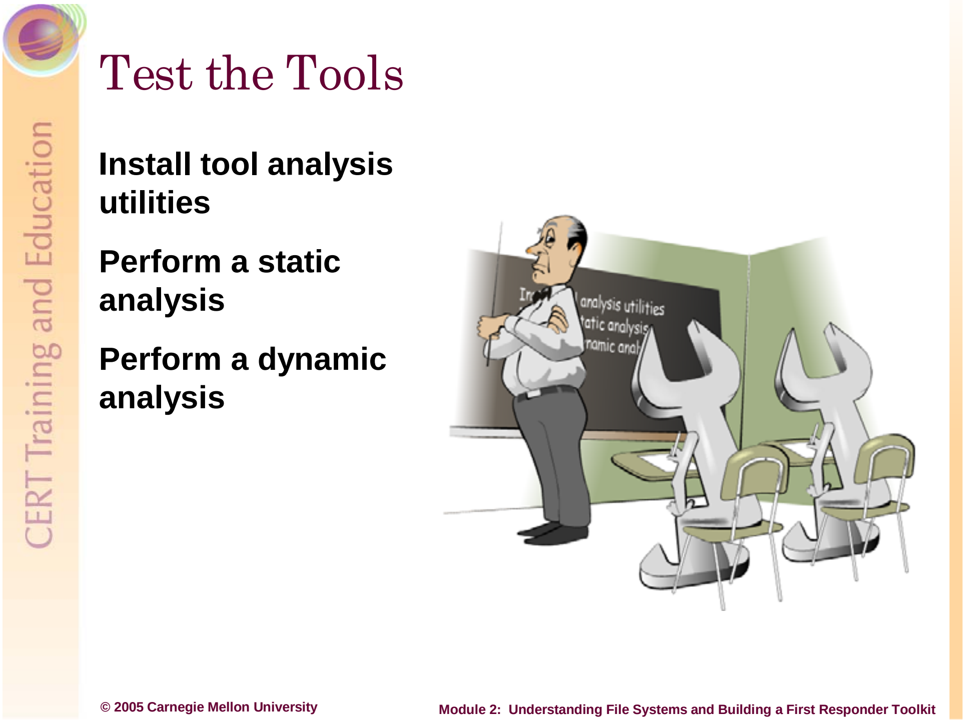
Any other items that are required for the tool to run (i.e., other executables, DLLs, and other files) should be copied onto the response disk. Shared libraries required for Linux commands should be copied to the response disk and the executable associated with the command should be statically compiled to ensure that the command does not reference the system as it is executed. The first responder identifies these required files and shared libraries during tool testing, the final step in this first responder toolkit methodology.

Determine how you will save the forensic tool output. You can save the output to the response disk or to another system on the network. The output should never be saved to the live suspicious computer.

Label your response disk properly with the following minimum information:

- time and date
- name of person who created and verified the response disk
- if the response disk contains any output files or collected evidence

The National Institute of Justice (NIJ) provides guidelines for documenting computer evidence [NIJ 04].



CERT Training and Education

Test the Tools

- Install tool analysis utilities
- Perform a static analysis
- Perform a dynamic analysis

analysis utilities
tatic analysis
namic anal

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3.3.4 Test the Tools

Now that you have identified and profiled forensic tools, you are ready to test each tool on the forensic testbed. During testing, the first responder can examine tool performance, functionality, and tool generated output and—most importantly—determine what affects the tool may have on a typical system.

Utilities like *Regmon*, *Filemon*, *ListDLLs*, *Dependency*, *LANguard* or *Tripwire* determine each tool’s impact on the system. They allow the first responder to monitor and catch changes or references to the forensic testbed system caused by the forensic tools and provide output that can be used to document the tools.

Install Tool Analysis Utilities

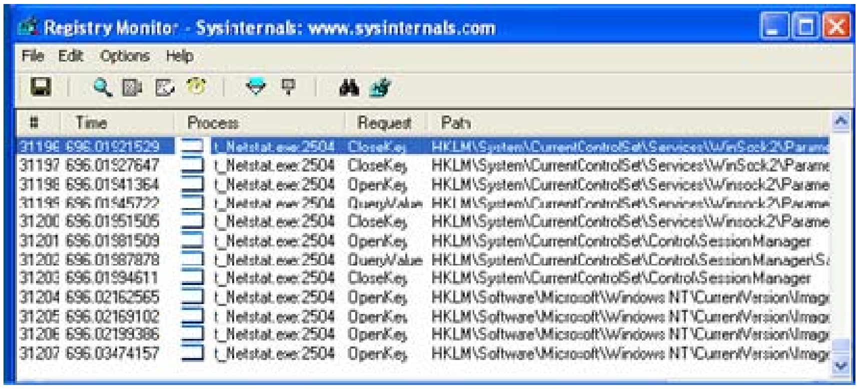
Acquire, unzip, and install the following utilities on the testbed system.

Filemon (Windows, Linux): Open and configure *Filemon*.

Filemon monitors and displays file system activity on a system in real time. Its advanced capabilities make it a powerful tool for exploring the way Windows works, seeing how applications use the files and DLLs, or tracking down problems in system or application file configurations. *Filemon's* time-stamping feature shows you precisely when every open, read, write, or delete occurs, and its status column displays the outcome. It begins monitoring when you start it, and its output window can be saved to a file for offline viewing. It has full search capability that can be filtered if you experience information overload [Russinovich 04d].

<u>*Regmon* (Windows)</u>: Open and configure *Regmon*.

Regmon monitors the registry and shows you—in real time—which applications are accessing your registry, which keys the tools are accessing, and the registry data that they are reading and writing. With static registry tools, you might be able to see which registry values and keys changed. With *Regmon*, you see which values and keys are changed and how [Russinovich 00].



#	Time	Process	Request	Path
31196	696.01921529	t_Netstat.exe:2504	CloseKey	HKLM\System\CurrentControlSet\Services\WinSock2\Parame
31197	696.01927647	t_Netstat.exe:2504	CloseKey	HKLM\System\CurrentControlSet\Services\WinSock2\Parame
31198	696.01941364	t_Netstat.exe:2504	OpenKey	HKLM\System\CurrentControlSet\Services\Winsock2\Parame
31199	696.01945722	t_Netstat.exe:2504	QueryValue	HKLM\System\CurrentControlSet\Services\Winsock2\Parame
31200	696.01951505	t_Netstat.exe:2504	CloseKey	HKLM\System\CurrentControlSet\Services\Winsock2\Parame
31201	696.01981509	t_Netstat.exe:2504	OpenKey	HKLM\System\CurrentControlSet\Control\SessionManager
31202	696.01987878	t_Netstat.exe:2504	QueryValue	HKLM\System\CurrentControlSet\Control\SessionManager\S
31203	696.01994611	t_Netstat.exe:2504	CloseKey	HKLM\System\CurrentControlSet\Control\SessionManager
31204	696.02162565	t_Netstat.exe:2504	OpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Imag
31205	696.02169102	t_Netstat.exe:2504	OpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Imag
31206	696.02199386	t_Netstat.exe:2504	OpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Imag
31207	696.03474157	t_Netstat.exe:2504	OpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Imag

Figure 7: A Regmon *Listing*

<u>*ListDLLs* (Windows)</u>: Run ListDDLs.

ListDLLs shows you the full path names of loaded DLL modules. In addition, ListDLLs will flag loaded DLLs that have different version numbers than their corresponding on-disk files (which occurs when the file is updated after a program loads the DLL), and can tell you which DLLs were relocated because they are not loaded at their base address [Russinovich 00].

<u>*Dependency* (Windows)</u>: Run depends.exe.

Dependency Walker is a free utility that builds a hierarchical tree diagram of all dependent modules. For each module, it lists exported functions and functions called by other modules. Another view displays the minimum set of required files and details about each file.

For more details, visit http://www.dependencywalker.com.

LANguard (Windows) and *Tripwire* (Linux): Set up, configure, and run *Tripwire* or *LANguard SIM*.

LANguard and *Tripwire* monitor file system integrity. These utilities will prove to be useful in seeing if the tools change the MAC times of any files in the file system. If the tool that you are testing does in fact touch the file system during your testing it is extremely important to document this and see if the changes were significant or malicious.

Strace.exe (Windows): *Strace* executes a program, and optionally the children of the program, *stdout*. Use the –o option to send output to a file. Use the -w option to start a *strace* session in a new window. This is useful for sessions that take a long time to complete. It is meant to be used like the *strace* (or *truss*) on Linux.

Strace for Windows systems is still in a Beta status and should be used with caution.

For more details, visit http://www.bindview.com/Support/RAZOR/Utilities/Windows/.

Exetype.exe (Windows, Linux): *Exetype* is a command-line utility available in the Windows Resource Kit. It is used to find the executable type and determine the OS and processor required to run the program file. It is useful when dealing with an unknown program.

Perform a Static Analysis of the Tools

Static and dynamic analyses allow you to disprove an initial assumption that each forensic tool is malicious. In a static analysis, you review each tool's documentation and file types, perform string searches, and view binary data (the executable) using a hex editor. In the process, a better understanding of the functionality, trustworthiness, and reliability of each tool is gained.

Tool Documentation

Review the tool's documentation to determine its functionality, supporting commands, and proposed system affects.

Determine the File Type

Determine the file type using the Linux *file* command or the Windows Resource Kit *exetype* command. This verifies that the correct file type is reflected in the downloaded tool's filename extension.

Display Legible Strings

Display legible strings, such as ASCII and Unicode strings, within the binary using the Linux or Windows command strings. Look at the keywords, command-line arguments, and variables to understand the tool's purpose [Mandia 01].

The *strings* command in Linux displays all humanly legible strings longer than four characters. If the programmer was verbose, these strings can reveal function names, variable names, defined strings, and—if you are lucky—complete pathnames of the source code used to build the executable.

Always use the -a flag, as shown below, to process the entire file.

```
strings -a <filename> | less
```

View Binary Data

View binary data to uncover abnormal function names. The rogue code’s names may provide clues about the function’s purpose.

For Linux systems, use the GNU debugger *gbd* with info variables as shown below.

```
gbd <filename> info variable
```

For Windows systems, use any freeware hex editor.

Perform a Dynamic Analysis of the Tools

In a dynamic analysis, you use traces to determine file access and dependencies. In the process, you can determine the total system impact of each tool.

Document and save all results.

Kernel Access

S*trace* is a Linux utility that wiretaps the system and reports every access of kernel resources. Whenever an application or tool writes to the screen, performs IO with network, disk, or other devices, or allocates memory, *strace* tracks it. Consider performing string searches on the outputted file, which can be difficult to read [Mandia 01].

Exetype is a Windows Resource Kit utility and a Linux command. *Exetype* is a power tool for extracting information about a running executable.

File System and Registry Key References

Use *Filemon* and *Regmon* to trace each tool’s file system and registry key references during execution. Before you begin, set the utility’s filters to look for the filename of the forensic tool’s executable only so that references by all other tools and application are ignored.

Review the *Filemon* and *Regmon* filter displays to see if your tool references any part of the file system outside of its own location (i.e., CD-ROM or thumb drive), and then document the output. For outside references, document the target and the time it was referenced.

DLL/Shared Library Dependencies

The following utilities identify which files each forensic file references, which allows you to rank each tool’s impact on a system and understand the expected side affects of each tool.

Windows: Use *ListDLLs* or *Dependency* to determine DLL dependencies.

For *Dependency*, use the Open button to navigate to and open your tool. *Dependency* then displays all DLLs required for the tool to run.

For *ListDLLs*, run your forensic tool first and then run *ListDLLs* to display the loaded DLLs.

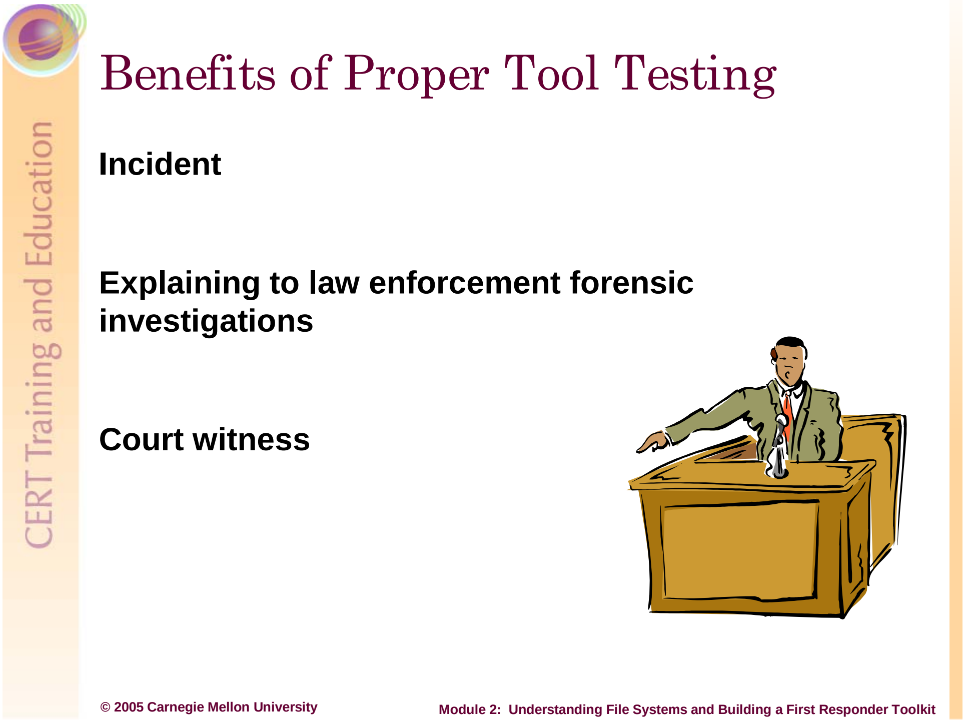
Linux: Use the *ldd* command to determine shared library dependencies. The *ldd* command prints the shared libraries required by each program or command specified on the command line.

Create a First Responder Toolkit Logbook

After testing, create a folder or file that includes the following for each forensic tool:

- the profile
- documentation
- static analysis results
- dynamic analysis results

This logbook should be kept handy. During the actual data collection described in Modules 3 and 4, you will refer to this book to determine which tools best fit your system and situation.



Benefits of Proper Tool Testing

Incident

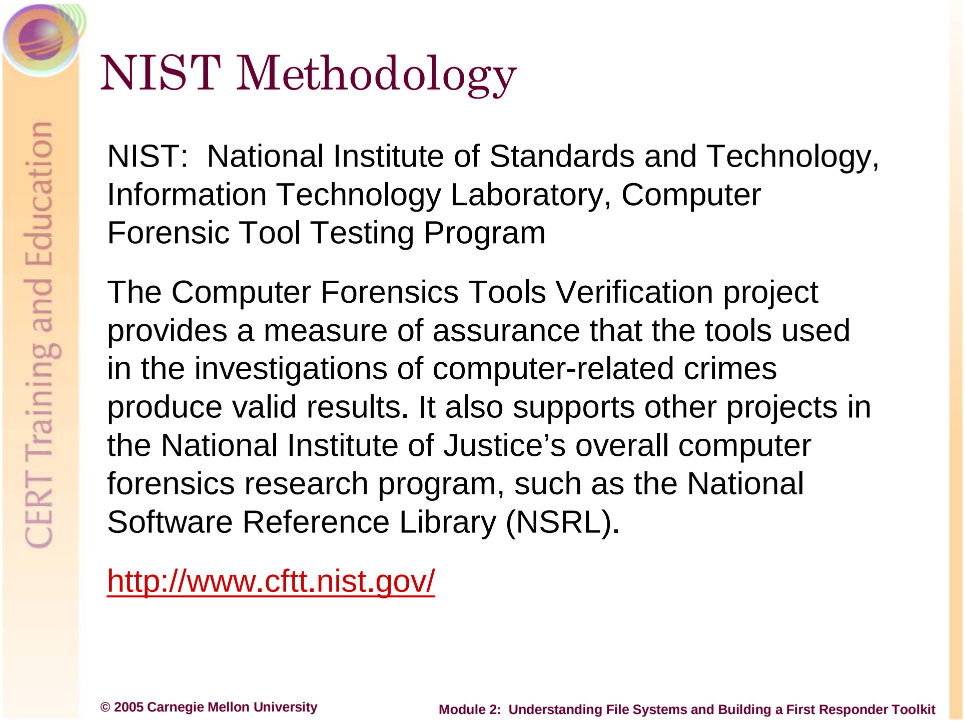
Explaining to law enforcement forensic investigations

Court witness

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3.3.5 Benefits of Proper Tool Testing

If you follow these best practices for testing tools before you use them, you can significantly reduce the time it takes to respond to a security incident while establishing the reliability of each tool’s output for use in a forensic investigation or court of law.


CERT Training and Education

NIST Methodology

NIST: National Institute of Standards and Technology, Information Technology Laboratory, Computer Forensic Tool Testing Program

The Computer Forensics Tools Verification project provides a measure of assurance that the tools used in the investigations of computer-related crimes produce valid results. It also supports other projects in the National Institute of Justice’s overall computer forensics research program, such as the National Software Reference Library (NSRL).

http://www.cftt.nist.gov/

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.3.3.6 NIST Methodology

Before we developed our methodology for creating a first responder toolkit, we examined the National Institute of Standards and Technology (NIST) Computer Forensics Tool Testing Program [NIST 03]. The following is an excerpt.

> *CFTT Methodology Overview*
>
> *The testing methodology developed by NIST is functionality driven. The activities of forensic investigations are separated into discrete functions or categories, such as hard disk write protection, disk imaging, string searching, etc. A test methodology is then developed for each category. Currently we have developed a methodology for disk imaging tools and are developing a methodology for software hard disk write blocking tools. Deleted file recovery tools will be the next category for development of a test methodology.*
>
> *The CFTT testing process is directed by a steering committee composed of representatives of the law enforcement community. Included are the FBI, DoD, NIJ (representing state and local agencies), NIST/OLES and other agencies. Currently the steering selects tool categories for investigation and tools within a category for actual testing by CFTT staff. A vendor may request testing of a tool, however the steering committee makes the decision about which tools to test.*
>
> *Under the disk imaging category the tools selected initially for testing were: Linux **dd**, and **SafeBack**. The **RCMP hdl** was selected for the hard disk write block category. Final test reports are posted to a Web site maintained by NIJ.*

1. Specification development process

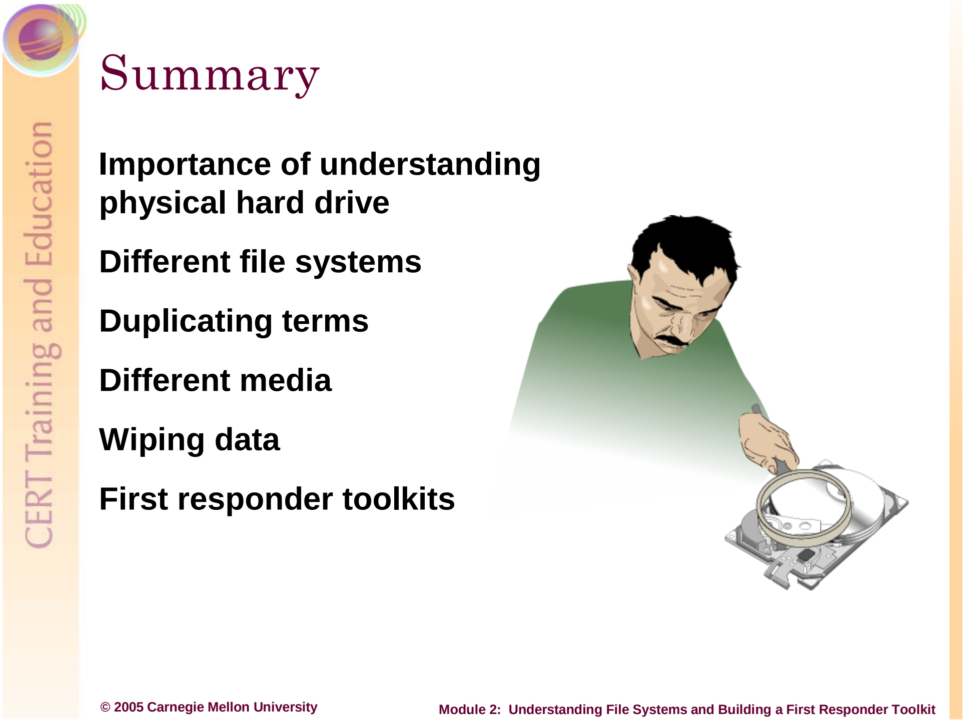
After a tool category and at least one tool is selected by the steering committee the development process is as follows:

1. *NIST and law enforcement staff develops a requirements, assertions and test cases document (called the tool category specification).*
2. *The tool category specification is posted to the Web for peer review by members of the computer forensics community and for public comment by other interested parties.*
3. *Relevant comments and feedback are incorporated into the specification.*
4. *A test environment is designed for the tool category.*

2. Tool test process

After a category specification has been developed and a tool selected, the test process is as follows:

1. *NIST acquires the tool to be tested.*
2. *NIST reviews the tool documentation.*
3. *NIST selects relevant test cases depending on features supported by the tool.*
4. *NIST develops test strategy.*
5. *NIST executes tests.*
6. *NIST produces test report.*
7. *Steering Committee reviews test report.*
8. *Vendor reviews test report.*
9. *NIST posts support software to Web.*
10. *NIJ posts test report to Web.*



CERT Training and Education

Summary

Importance of understanding physical hard drive

Different file systems

Duplicating terms

Different media

Wiping data

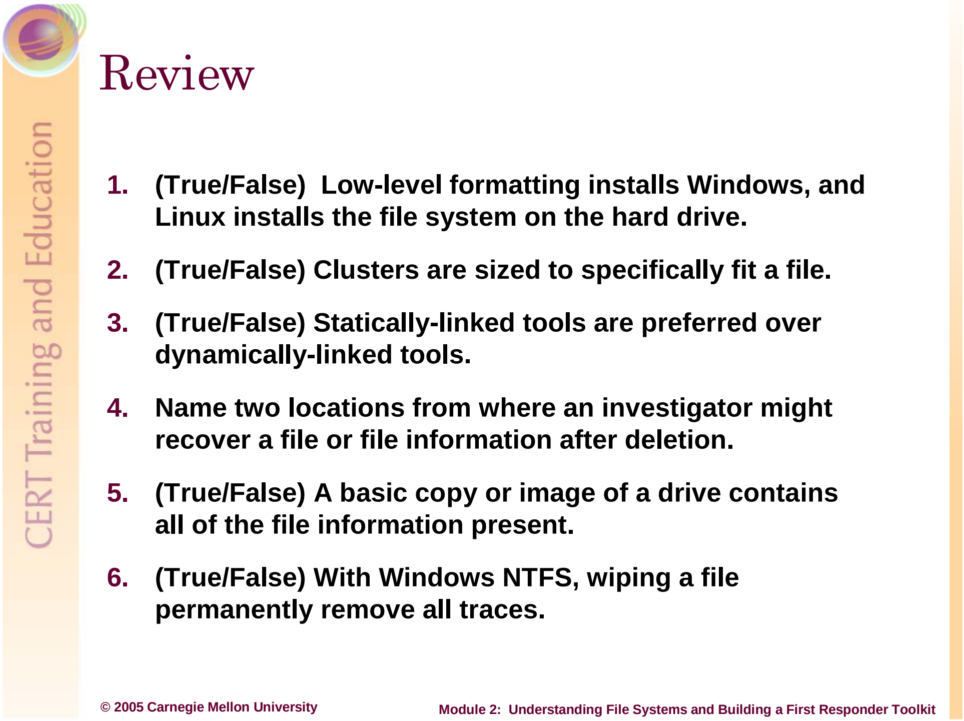
First responder toolkits

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.4 Summary

From this module you have been exposed to some basic hard drive geometry, and an overview of how the hard drive works with an OS's file system to store data. You should understand that files leave traces when deleted and know that those file artifacts are hidden in swap and slack space. Together these provide for a better understanding of the fragile nature of forensic evidence.

Building a first responder toolkit using this module's methodology provides you with a baseline understanding of the different systems in the work environment. Establishing and using this repeatable, defined collection process is essential if, based on collected evidence, you take administrative action against an individual or submit evidence to a court of law. Without a repeatable, defined process, your evidence will be challenged. Investing the time and effort in creating a first responder toolkit pays off when an incident occurs and the scramble to collect evidence ensues.



CERT Training and Education

Review

1. (True/False) Low-level formatting installs Windows, and Linux installs the file system on the hard drive.
2. (True/False) Clusters are sized to specifically fit a file.
3. (True/False) Statically-linked tools are preferred over dynamically-linked tools.
4. Name two locations from where an investigator might recover a file or file information after deletion.
5. (True/False) A basic copy or image of a drive contains all of the file information present.
6. (True/False) With Windows NTFS, wiping a file permanently remove all traces.

© 2005 Carnegie Mellon University

Module 2: Understanding File Systems and Building a First Responder Toolkit

2.5 Review

1. False. Low-level formatting occurs at the factory and installs the sectors and tracks on the disk. High-level formatting establishes the OS’s file system.

2. False. Clusters are a fixed size with incorporate several sectors. Their exact size is determined by the Windows version and size of the hard drive.

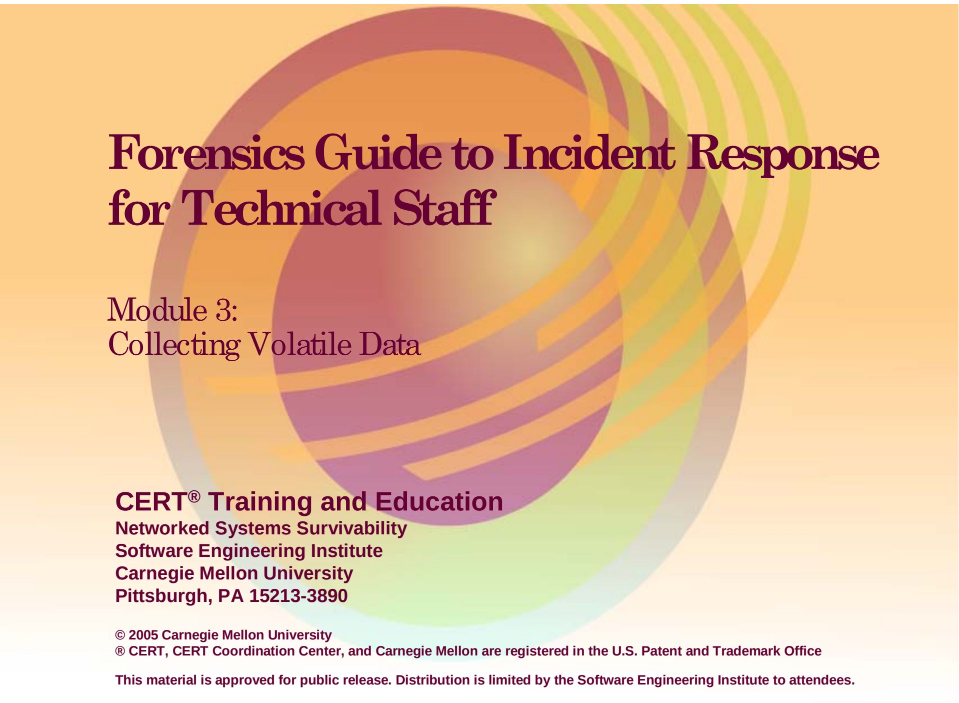
3. True. Statically-linked is preferred because all file dependencies are trusted and not accessed from the suspicious computer’s hard drive.

4. Swap space (virtual memory), slack space, unallocated space, windows artifacts (i.e., multiple data streams, temp files, registry, shortcuts).

5. False. Only a bit-by-bit or duplicate image provides all binary data from a disk. Using the basic copy command or imaging only reproduces the allocated space.

6. False. Windows artifacts and alternate data streams in the NTFS hold keys to a file’s existence even after a DoD standard wipe.

3 Module 3: Collecting Volatile Data



Forensics Guide to Incident Response for Technical Staff

Module 3:
Collecting Volatile Data

CERT® Training and Education
Networked Systems Survivability
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

This material is approved for public release. Distribution is limited by the Software Engineering Institute to attendees.

3.1 Introduction

This module briefly reviews some best practices, techniques, and tools for collecting volatile data from live Windows and Linux systems. It also explains the importance of collecting volatile data before it is lost or changed on the suspicious computer.

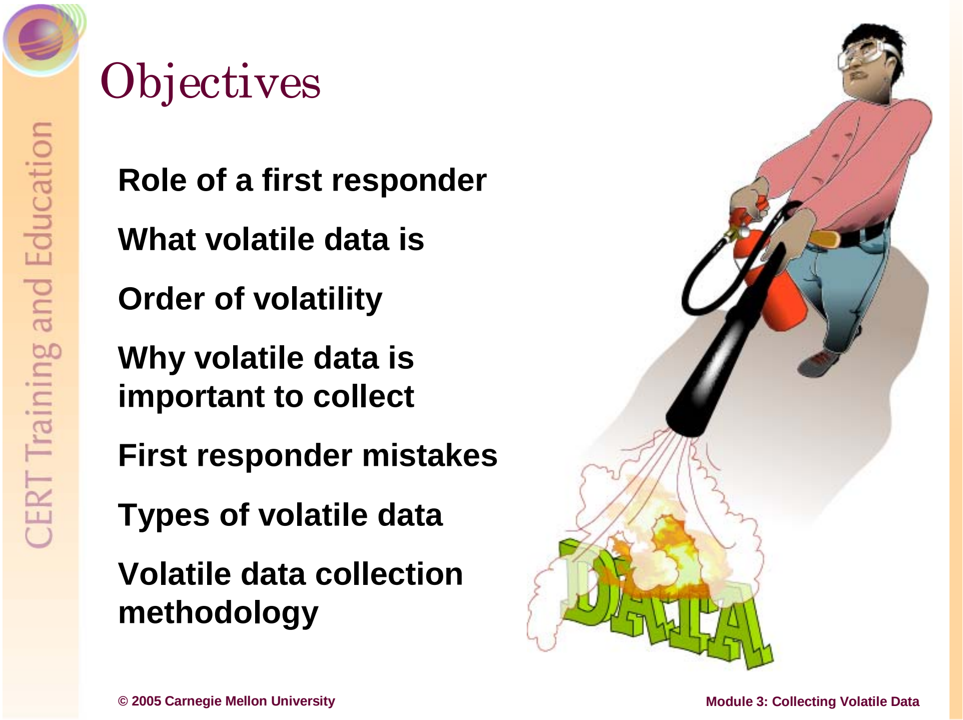
Traditionally, computer forensics has focused on researching, developing, and implementing proper techniques, tools, and methodologies to collect, store, and preserve sensitive data that is left on a system's hard drive(s).

First responders (system and network administrators, law enforcement officers, and others) generally react to a computer security incident by properly shutting down and securing the suspicious computer. After shutdown, persistent data on the computer's hard drive or other persistent storage media devices is collected. But the shutdown precludes the collection of volatile data, critical sources of information that are lost when the computer is rebooted or shut down. To collect volatile data, the first responder needs a running computer system.

In recent years, law enforcement and information security professionals have invested significant resources into developing forensic data retrieval tools like Guidance Software's EnCase, Access Data's FTK, and NTI's Law Enforcement Suite. However, these tools and techniques focus almost exclusively on extracting and analyzing nonvolatile (persistent) data from hard drives. Little attention has been given to developing a portable, safe, all-in-one application for collecting volatile data from a live machine.

There are many open source tools that allow the first responder to pull volatile information from a live system. However, most open source tools are specifically designed to collect portions of volatile data, not the complete set of volatile data (e.g., Sysinternals' PsList generates details about processes that are currently running).

First responders need to identify and gather a set of open source tools (including native commands) that collect the range of volatile data. Then they can use scripting languages (e.g., Perl, VBScript, batch files, and shell scripts) to execute all tools and commands and automate the data collection.



CERT Training and Education

Objectives

- Role of a first responder
- What volatile data is
- Order of volatility
- Why volatile data is important to collect
- First responder mistakes
- Types of volatile data
- Volatile data collection methodology

© 2005 Carnegie Mellon University

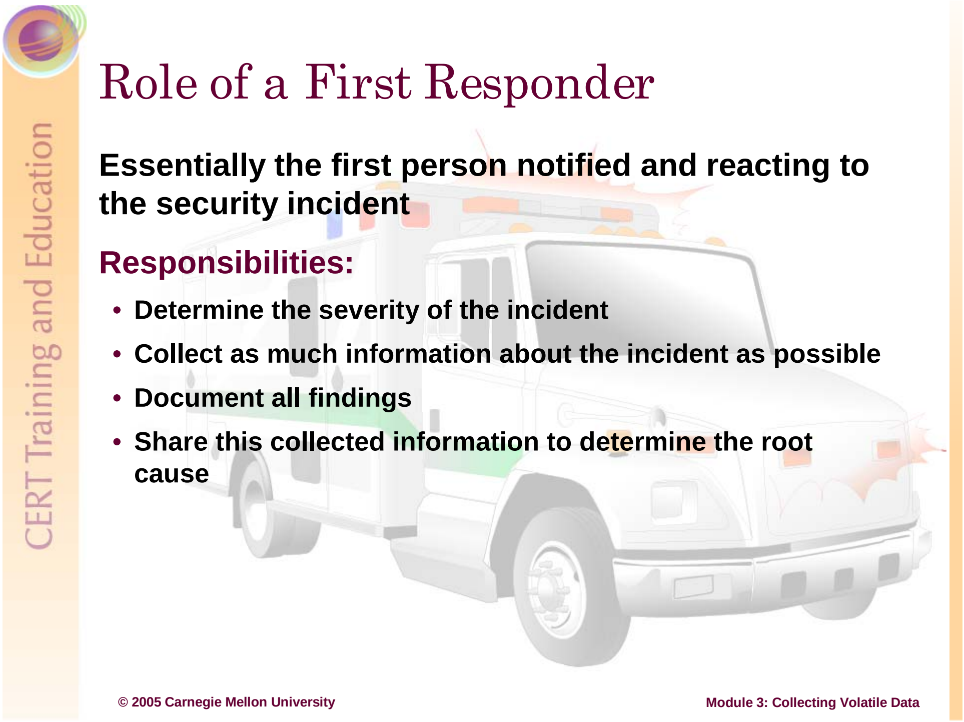
Module 3: Collecting Volatile Data

3.2 Objectives

This module focuses on the critical role of the first responder in collecting volatile data from a live suspicious machine during a security incident. Common first responder mistakes are also highlighted so that you do not hamstring your efforts.

Discussions of volatile data characteristics (e.g., what it is, the different types, the order of volatility, and why it should be collected) and the tools used to collect it should give the first responder a working knowledge of the subject.

The bulk of this module introduces the methodology and specific tools for collecting volatile data. It includes steps for safely collecting volatile data so that you can determine the current state of the computer, the root cause of the incident, and what to do next.



CERT Training and Education

Role of a First Responder

Essentially the first person notified and reacting to the security incident

Responsibilities:

- **Determine the severity of the incident**
- **Collect as much information about the incident as possible**
- **Document all findings**
- **Share this collected information to determine the root cause**

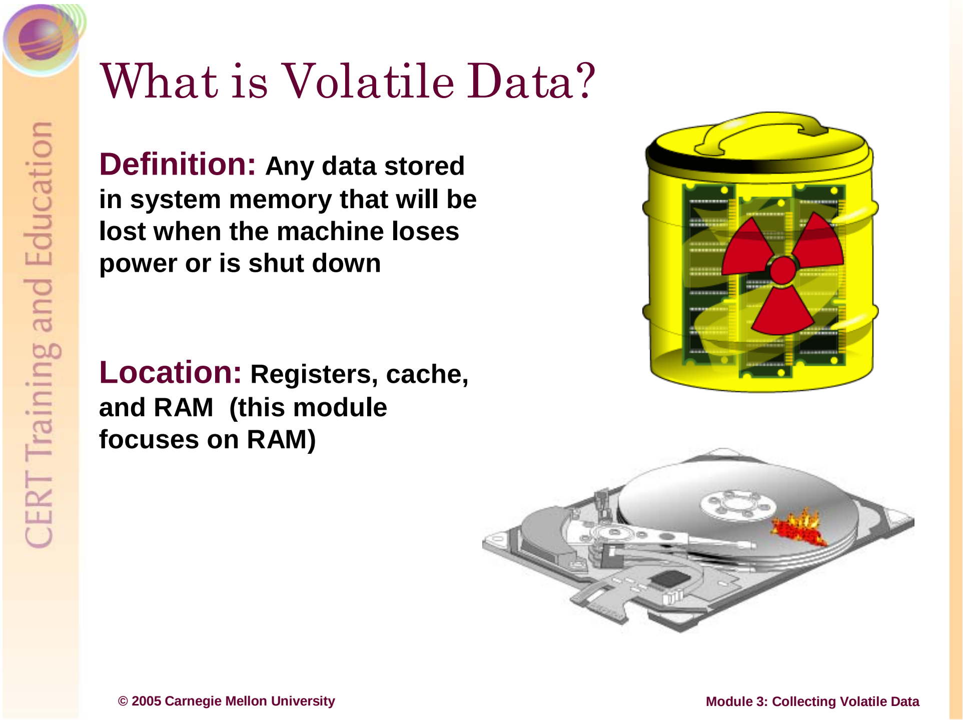
© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.3 Role of a First Responder

The first responder to a security incident has a unique and important position to play. He will generally be the first person notified and the first person to react. In most cases, the first responder is a system or network administrator, but by definition it is whoever is assigned to handle security incidents and determine their root causes.

The first responder should be prepared and his actions should be planned. Deliberate – rash or hurried actions could damage potential evidence. He should have a first responder toolkit and a predetermined incident response plan to follow regardless of the type of data (volatile, persistent, or both) being collected.



CERT Training and Education

What is Volatile Data?

Definition: Any data stored in system memory that will be lost when the machine loses power or is shut down

Location: Registers, cache, and RAM (this module focuses on RAM)

Module 3: Collecting Volatile Data

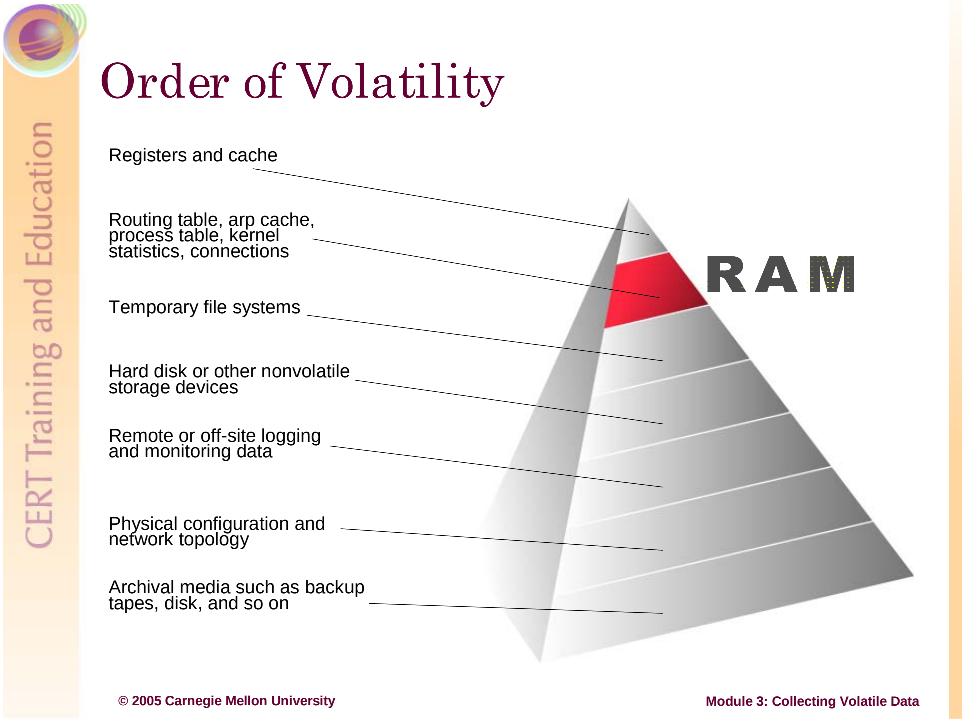
3.4 What is Volatile Data?

Volatile data is stored in system memory (e.g., system registers, cache, RAM) and is lost if the machine loses its power, is shut down, or rebooted. Volatile data collection focuses on collecting data (primarily from RAM) that could be lost if the computer is shut down or rebooted. Volatile data should be collected if you are not sure why a computer is acting abnormally, if you notice suspicious user activity or if you have been alerted that a rule or policy has been violated such as firewall or IDS alert. In any case, the first response to a computer security incident should be to collect volatile data and analyze the results to determine a next course of action.

Persistent data resides in the system's hard drives or other nonvolatile storage devices (e.g., connected USB drives, flash cards, CD-ROMs, and other external hard drives) and is typically not lost when the machine is shut down or rebooted. Generally, persistent data should be collected when it is clear that evidence related to the computer security incident resides in the persistent storage areas.

For both collection strategies, preventing contamination of the suspicious computer is an ongoing issue. For persistent data collection, contamination can be controlled by current techniques, tools, and methodologies. One such tool is *dd.exe*, a free imaging utility developed by George M. Garner Jr. that has the capability to create a bit-by-bit copy of the suspicious computer's hard drive. If you create MD5 checksums of the hard drive before and after using a utility like DD, you can compare them and authenticate the copy. For volatile data collection, contamination is harder to control because tools and commands may change file access dates and times, use shared libraries or DLLs, trigger the execution of malicious software (malware), or—worst case—force a reboot and lose all volatile data. These potential effects should have been uncovered and documented during Module 2's tool testing phase of creating a first responder toolkit.

Obviously, you can not make a bit-by-bit copy of a live computer. But if you use best practices and a first responder toolkit, you can use the collected data and a reproducible methodology to reconstruct a logical representation of the current state of the suspicious system.



CERT Training and Education

Order of Volatility

Registers and cache

Routing table, arp cache, process table, kernel statistics, connections

Temporary file systems

Hard disk or other nonvolatile storage devices

Remote or off-site logging and monitoring data

Physical configuration and network topology

Archival media such as backup tapes, disk, and so on

RAM

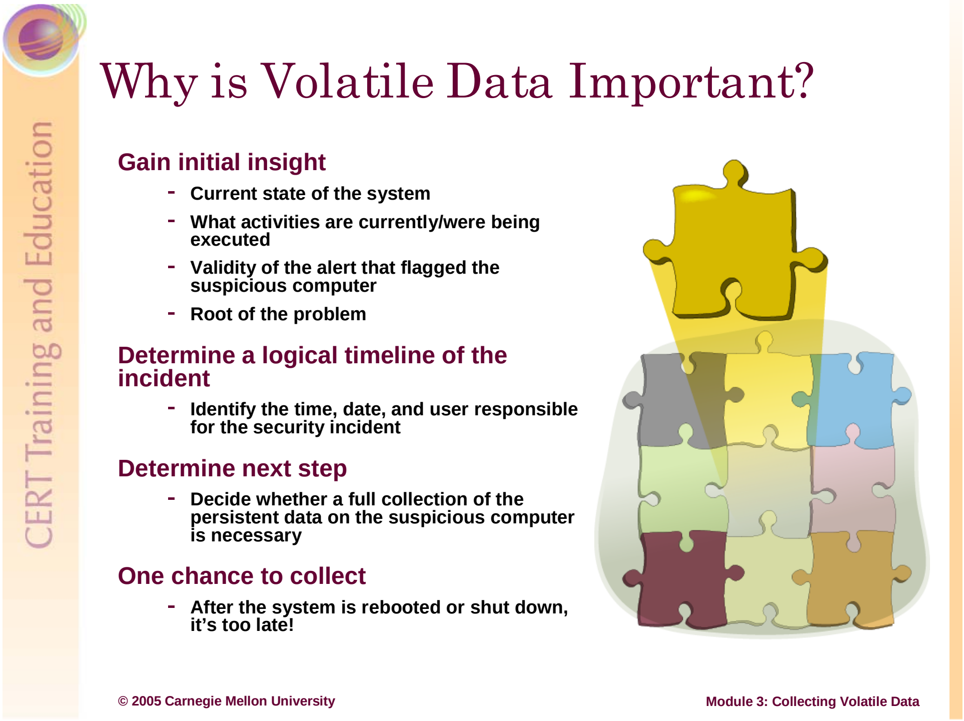
© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.5 Order of Volatility

As you collect data (i.e., potential evidence) from a live computer, consider the data's order of volatility: that is, you collect data that has the highest chance of being changed, modified, or lost first. The order of volatility for Windows and Linux computers is the same [Brezinski 02].

This module focuses on data stored in RAM. It shows how to collect this data safely and securely while minimizing the footprint you leave on the suspicious computer.


CERT Training and Education

Why is Volatile Data Important?

Gain initial insight

- Current state of the system
- What activities are currently/were being executed
- Validity of the alert that flagged the suspicious computer
- Root of the problem

Determine a logical timeline of the incident

- Identify the time, date, and user responsible for the security incident

Determine next step

- Decide whether a full collection of the persistent data on the suspicious computer is necessary

One chance to collect

- After the system is rebooted or shut down, it's too late!

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.6 Why is Volatile Data Important?

Volatile data stored in RAM must be obtained immediately after the security incident is reported so that data about the current state of the computer—including logged on users, running processes, and open connections—can be collected. This data helps you determine a logical timeline of the security incident and the possible users responsible.

After you collect as much volatile data as you can, decide on the next step. For example, you may have enough data to determine that the initial alert that flagged this computer is valid. You may then discover a rogue process, a likely root cause. And if the rogue process is running on a critical asset, you will have to remove it remotely and then verify if it is malicious.

The key is to view the suspicious computer as completely unreliable until—using the methodology and tools described in this module—some level of trust is established.

CERT Training and Education

Common First Responder Mistakes

Shutting down or rebooting the suspicious computer

Assuming that some parts of the suspicious machine may be reliable and usable

Not having access to baseline documentation about the suspicious computer

NOT DOCUMENTING THE DATA COLLECTION PROCESS

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.7 Common First Responder Mistakes

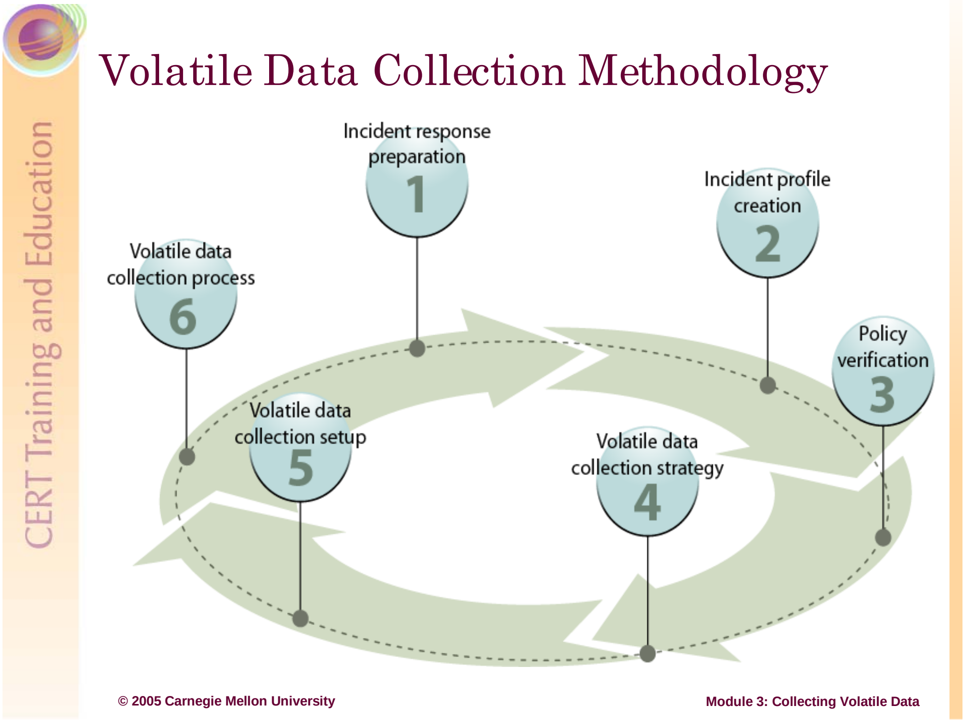
Because many first responders play dual roles as system and network administrators, they have little time to properly plan for and then handle security incidents. Swamped with routine day-to-day tasks, they may never consider developing a volatile data collection methodology. For the methodology to be successful, though, first responders should avoid the two most critical mistakes:

- shutting off or rebooting the machine

  In this case, all volatile data is lost: connections and running processes are closed and MAC times are changed. Closing an open shell or terminal may have the same effect.

- assuming that some parts of the suspicious computer are reliable

  In this case, using native commands on the suspicious computer may trigger Trojans, time bombs, and other malware to delete key volatile data.



CERT Training and Education

Volatile Data Collection Methodology

1 Incident response preparation

2 Incident profile creation

3 Policy verification

4 Volatile data collection strategy

5 Volatile data collection setup

6 Volatile data collection process

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.8 Volatile Data Collection Methodology

Before we introduce specific tools, we need to provide a sound volatile data collection methodology. Develop and use this methodology so that you collect volatile data the first time: you only get one chance.

Having a methodology is important. It provides a documented approach for performing activities in a coherent, consistent, accountable, and repeatable manner [ICH 03].

Harlan Carvey [Carvey 04] states that most system administrators do not have a clear process to follow when responding to security incidents. Often, an administrator’s default action, due to lack of knowledge or time, is to remove the OS and reload it from the original installation media. Several drawbacks are that a root cause analysis is never done and reloading the OS takes the system out of service. To avoid these drawbacks, an initial forensic collection of the volatile data on the live system should be performed.

What follows is a six part methodology for first responders to use when volatile data has to be collected from a live computer. The actual data collection is not performed until the final step.

CERT Training and Education

Step 1: Incident Response Preparation

Forensic Tool Test Bed

First responder toolkit

Creation of Collection policies

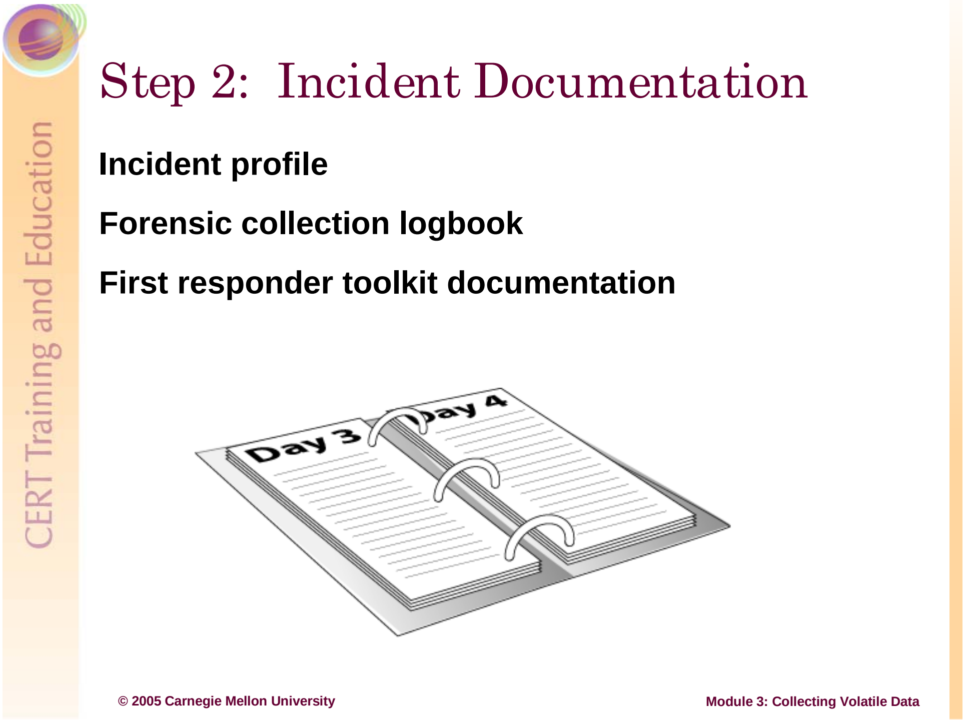
© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.8.1 Step 1: Incident Response Preparation

As a first responder, you can never eliminate or even anticipate every type of security incident or threat. You can, however, be prepared to react successfully and swiftly so that you can collect all types of volatile data from a live computer. Before an incident occurs, the following items should be in place.

- a first responder toolkit (response disk)
- an incident response team (IRT) or designated first responder (NIST's guidelines [NIST 01] are a good resource)
- forensic-related policies that allow for forensic collection



CERT Training and Education

Step 2: Incident Documentation

Incident profile

Forensic collection logbook

First responder toolkit documentation

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.8.2 Step 2: Incident Documentation

In your incident documentation, be straightforward and consistent so that your generated profiles and logs are organized and readable. For example, use consistent naming conventions for forensic tool output. Stamp your logs with the time, date, and identity of the person performing the forensic collection. These best practices provide an audit trail of the chain of custody for the security incident.

3.8.2.1 Incident Profile

Document as much information about the security incident as possible. The checklist shown in the graphic is taken from Foundstone's Incident Notification Checklist [Mandia 01]. Use the following profile information to determine your volatile collection strategy or next course of action.

- How was the incident detected?
- What is the scenario for the incident?
- What time did the incident occur?
- Who or what reported the incident?
- What hardware and software are involved?
- Who are the contacts for involved personnel?
- How critical is the suspicious computer?

3.8.2.2 Forensic Collection Logbook

During your forensic collection, use a logbook (i.e., a text file or notebook) to document all of your actions. The logbook provides a timeline of events and an audit trail if unforeseen things occur during the collection. Include the following:

- who is performing the forensic collection
- history of executed forensic tools and commands
- generated forensic tool and command output
- date and time of the executed commands and tools
- expected system changes or effects (i.e., changed MAC times for specific files) as first responder tools are executed (these effects were discovered in Module 2)

3.8.2.3 First Responder Toolkit Logbook

As you collect volatile data, refer to the first responder toolkit logbook that was created in Module 2. It contains information about forensic tool usage, output, and system affects for each collection tool or native command you tested. Use the logbook to determine which tools are the best for your situation.

Step 3: Policy Verification

Determine your authority to collect

Determine your manner to collect

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

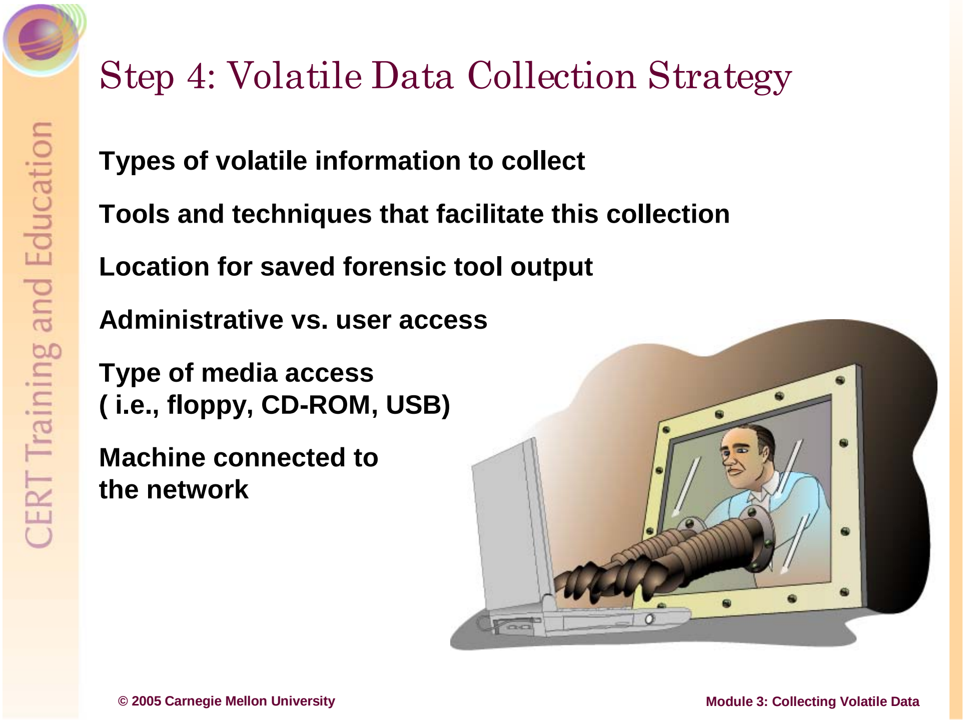
3.8.3 Step 3: Policy Verification

Make sure that any actions you plan to take as first responder do not violate any existing computer and network usage policies. In particular, make sure that you do not violate any rights of the registered owner or user of the suspicious computer.

- Examine any policies the user of the suspicious computer has signed.
- Determine if the user consented to monitoring either through a signed policy statement or through system banners.
- As necessary, determine the legal rights (including a review of federal statutes) that the user has so that you can determine what forensic capabilities and limitations you have. (See Module 1.)

We recommend that first responders read the following CERT Coordination Center publications before collecting any data:

- Establish Policies and Procedures for Responding to Intrusions
  http://www.cert.org/security-improvement/practices/p044.html
- CERT® Advisory CA-1992-19 Keystroke Logging Banner
  http://www.cert.org/advisories/CA-1992-19.html
- Steps for Recovering from a UNIX or NT System Compromise
  http://www.cert.org/tech_tips/win-UNIX-system_compromise.html#A.1



CERT Training and Education

Step 4: Volatile Data Collection Strategy

Types of volatile information to collect

Tools and techniques that facilitate this collection

Location for saved forensic tool output

Administrative vs. user access

Type of media access
(i.e., floppy, CD-ROM, USB)

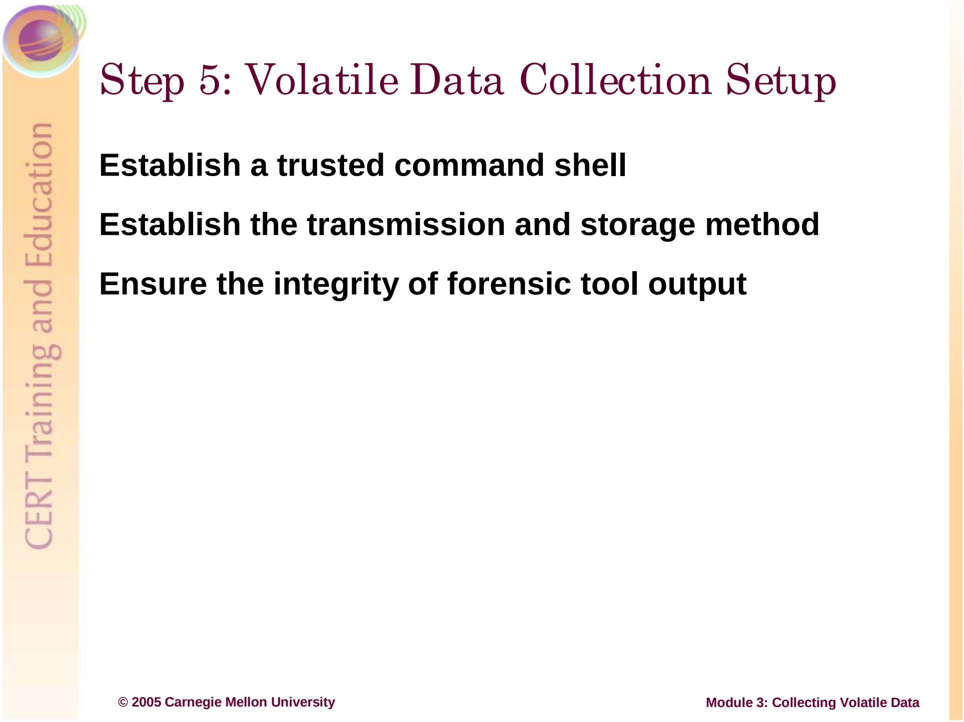
Machine connected to
the network

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.8.4 Step 4: Volatile Data Collection Strategy

No two security incidents will be the same. Use the first responder toolkit logbook and the questions from the graphic to develop a collection strategy that will address your situation and leave the smallest possible footprint on the suspicious computer.



CERT Training and Education

Step 5: Volatile Data Collection Setup

Establish a trusted command shell

Establish the transmission and storage method

Ensure the integrity of forensic tool output

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.8.5 Step 5: Volatile Data Collection Setup

3.8.5.1 Establish a Trusted Command Shell

As stated earlier, you do not want to make that common first responder mistake of trusting any parts of the suspicious computer. Therefore, do not open or use a terminal or command shell from the suspicious computer. This will minimize the footprint on the suspicious computer and prevent the triggering of any kind of malware that have been installed on the system.

The response disk that was developed during Module 2 should include a trusted command shell along with the forensic tools.

If the suspicious computer is a Linux system running X-Windows, exit X-Windows. Some UNIX flavors have a vulnerability that allows attackers to log your keystrokes [Mandia 01].

3.8.5.2 Establish a Method for Transmitting and Storing the Collected Information

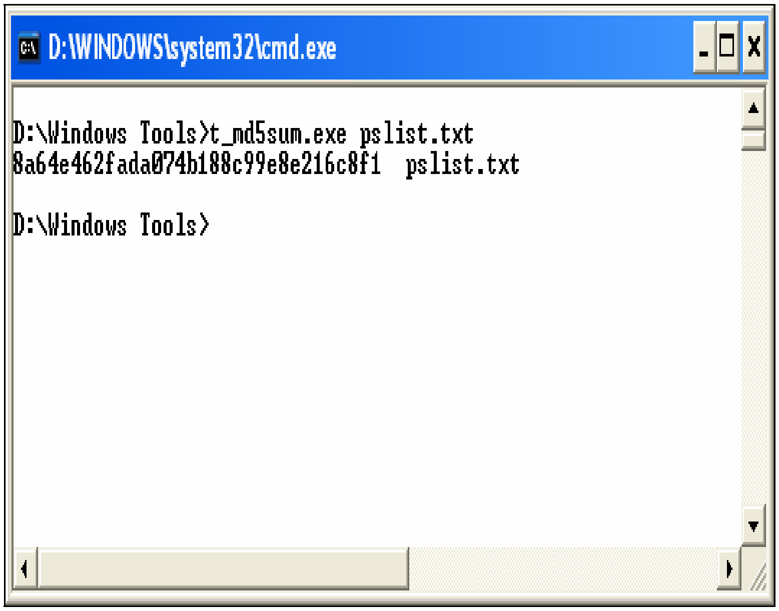
Because you may not have enough space on your response disk (i.e., USB drive, floppy, etc.) to collect forensic tool output, identify and document how that data will be transmitted from the live suspicious computer to the remote data collection system. Two reliable freeware utilities that transmit data remotely via a network are *Netcat* and C*ryptcat*.

Netcat is a networking utility that reads and writes data across network connections using the TCP/IP protocol. It is designed as a reliable "back-end" tool that can be used directly or driven by other programs and scripts. It is also a feature-rich network debugging and exploration tool, able to create almost any kind of connection you need and with several interesting built-in capabilities [GNU 04].

Cryptcat is a lightweight version of *netcat* with integrated transport encryption capabilities. It ensures data confidentiality while transmitting it over networks and is particularly useful if you suspect a network sniffer.

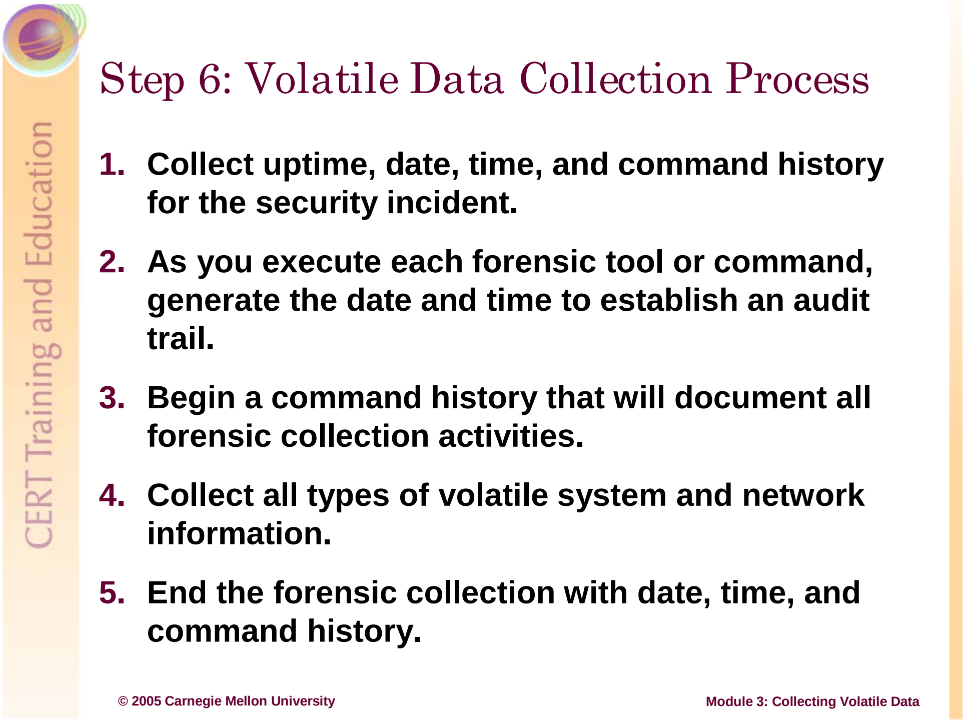
3.8.5.3 Ensure the Integrity and Admissibility of the Forensic Tool Output

To ensure the integrity and admissibility of forensic tool output, compute an MD5 hash of it. When an electronic file is transferred or sent to another party, a rehash verifies its integrity.



```
D:\Windows Tools>t_md5sum.exe pslist.txt
8a64e462fada074b188c99e8e216c8f1  pslist.txt

D:\Windows Tools>
```

Figure 8: An MD5 Hash



CERT Training and Education

Step 6: Volatile Data Collection Process

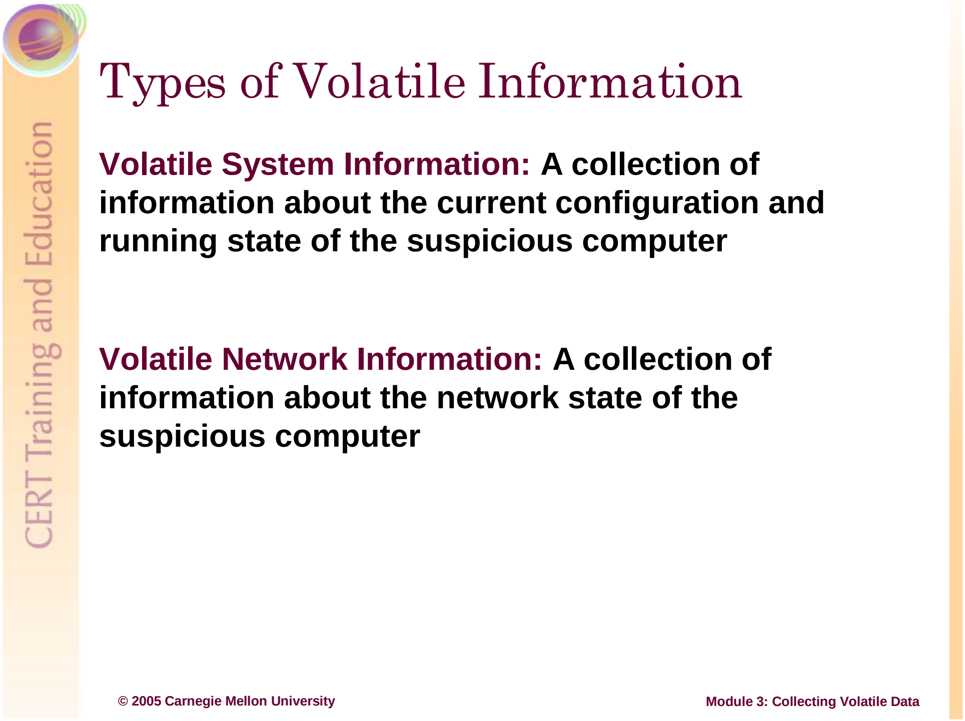
1. **Collect uptime, date, time, and command history for the security incident.**
2. **As you execute each forensic tool or command, generate the date and time to establish an audit trail.**
3. **Begin a command history that will document all forensic collection activities.**
4. **Collect all types of volatile system and network information.**
5. **End the forensic collection with date, time, and command history.**

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.8.6 Step 6: Volatile Data Collection Process

Now you are ready to collect volatile data. The remaining sections of this module describe each data type and specific tools for collecting it.


CERT Training and Education

Types of Volatile Information

Volatile System Information: **A collection of information about the current configuration and running state of the suspicious computer**

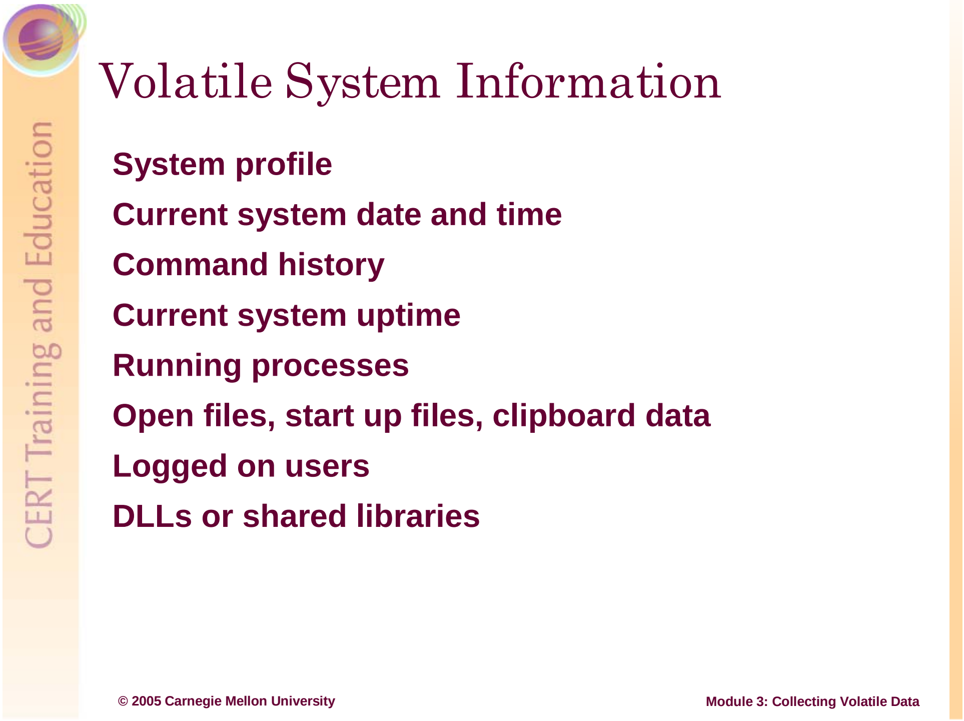
Volatile Network Information: **A collection of information about the network state of the suspicious computer**

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.9 Types of Volatile Information

Two types of volatile data that can be collected from the suspicious computer: system information and network information. The forensic collection tools that are described in the balance of this module are organized by these two data types.


CERT Training and Education

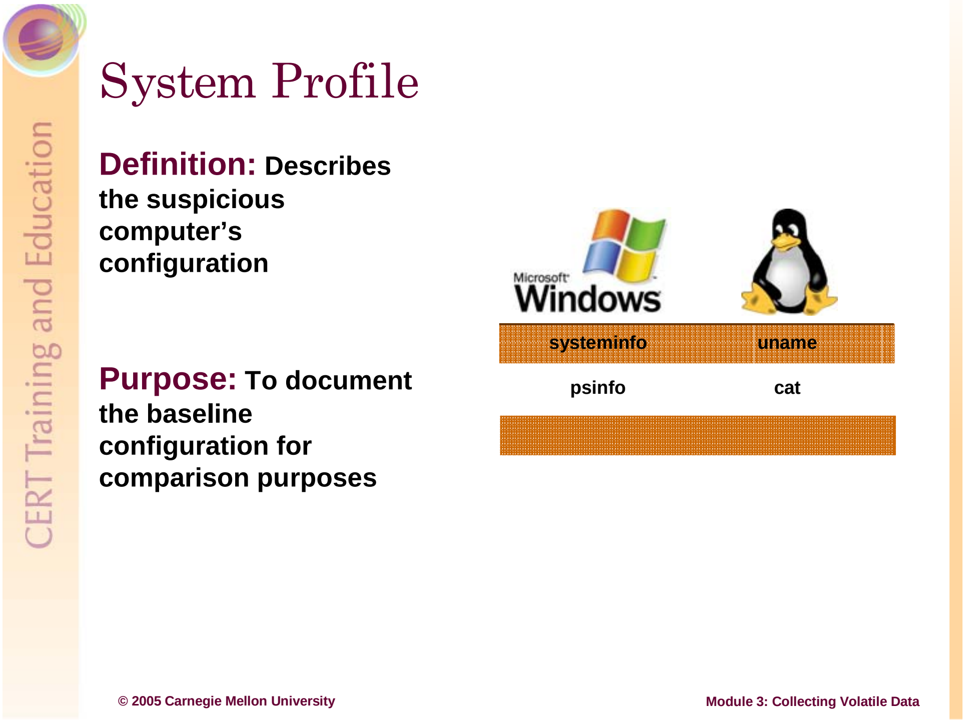
Volatile System Information

- System profile
- Current system date and time
- Command history
- Current system uptime
- Running processes
- Open files, start up files, clipboard data
- Logged on users
- DLLs or shared libraries

Module 3: Collecting Volatile Data

3.9.1 Volatile System Information

This list of volatile system information types, while not exhaustive, reveals enough details about the suspicious computer’s current state, configuration, and users for you to begin to understand the severity, probable causes, and perpetrator of the security incident.



CERT Training and Education

System Profile

Definition: Describes the suspicious computer’s configuration

Purpose: To document the baseline configuration for comparison purposes

Microsoft Windows	
systeminfo	uname
psinfo	cat

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.9.1.1 System Profile

The system profile describes the baseline configuration of the suspicious computer. Ideally, system and network administrators create these profiles for all computers within their IT and network infrastructures, but if one doesn’t already exist for the suspicious computer, create it now. This profile provides a physical “snapshot” of the computer and is often requested by a forensic examiner.

The system profile includes the following details about the configuration of the suspicious computer:

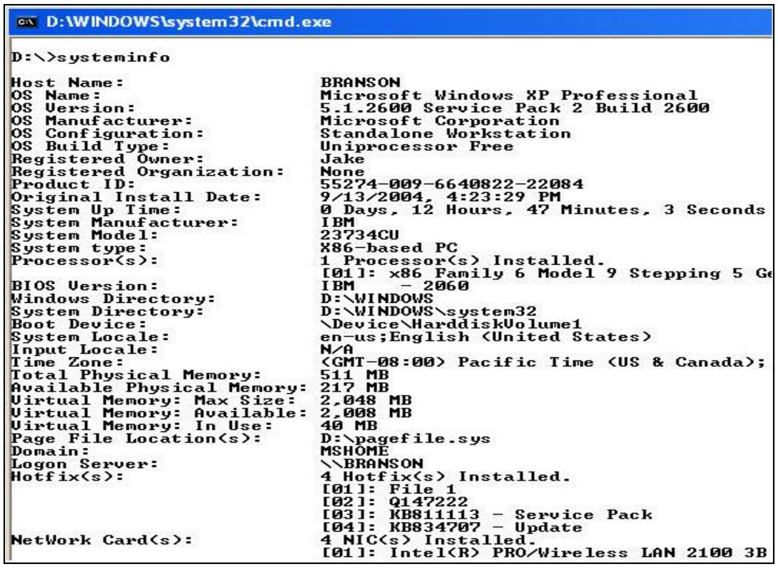
- OS type and version
- system installation date
- registered owner
- system directory
- total amount of physical memory
- pagefile location
- installed physical hardware and configurations
- installed software applications

The following sections describe tools and commands for collecting this information.

Systeminfo.exe (Windows)

The native *systeminfo.exe* command allows you to create a system profile that includes

- registered owner
- original install date
- system uptime
- BIOS version
- system directory
- logon server
- number of network cards installed



```
D:\>systeminfo

Host Name:                 BRANSON
OS Name:                   Microsoft Windows XP Professional
OS Version:                5.1.2600 Service Pack 2 Build 2600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Uniprocessor Free
Registered Owner:          Jake
Registered Organization:   None
Product ID:                55274-009-6640822-22084
Original Install Date:     9/13/2004, 4:23:29 PM
System Up Time:            0 Days, 12 Hours, 47 Minutes, 3 Seconds
System Manufacturer:       IBM
System Model:              23734CU
System type:               X86-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: x86 Family 6 Model 9 Stepping 5 Ge
BIOS Version:              IBM    - 2060
Windows Directory:         D:\WINDOWS
System Directory:          D:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              N/A
Time Zone:                 (GMT-08:00) Pacific Time (US & Canada);
Total Physical Memory:     511 MB
Available Physical Memory: 217 MB
Virtual Memory: Max Size:  2,048 MB
Virtual Memory: Available: 2,008 MB
Virtual Memory: In Use:    40 MB
Page File Location(s):     D:\pagefile.sys
Domain:                    MSHOME
Logon Server:              \\BRANSON
Hotfix(s):                 4 Hotfix(s) Installed.
                           [01]: File 1
                           [02]: Q147222
                           [03]: KB811113 - Service Pack
                           [04]: KB834707 - Update
NetWork Card(s):           4 NIC(s) Installed.
                           [01]: Intel(R) PRO/Wireless LAN 2100 3B
```

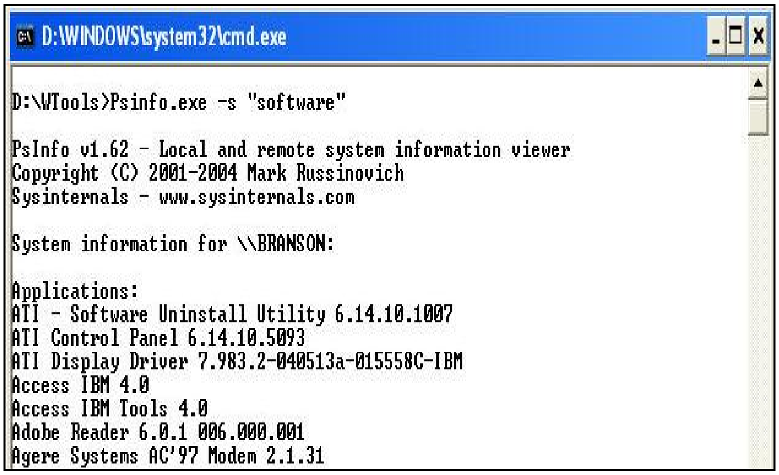
Figure 9: The systeminfo *Command*

PsInfo (Windows)

Sysinternals' *PsInfo* also allows you to create a system profile to include installed software packages and hot fixes, which the Systeminfo.exe command does not show you. By default, the PSInfo creates profiles of the local system. You can run it remotely by specifying a username and password of the suspicious computer.

PsInfo relies on remote registry access to obtain data on non-local systems. The remote system must be running the Remote Registry service and the account from which you run *PsInfo* must have access to the HKLM\System portion of the remote registry [Russinovich 04a].

PsInfo is versatile. Use it to query for types of installed software, hot fixes, hard drive size, and remaining unused space. In Figure 10, we used *Psinfo* to query the suspicious computer for the number of applications installed.

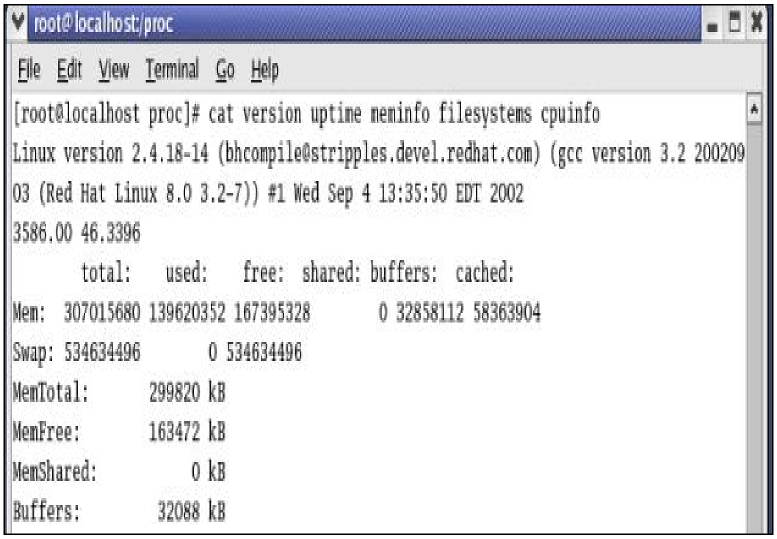


```
D:\WTools>Psinfo.exe -s "software"

PsInfo v1.62 - Local and remote system information viewer
Copyright (C) 2001-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

System information for \\BRANSON:

Applications:
ATI - Software Uninstall Utility 6.14.10.1007
ATI Control Panel 6.14.10.5093
ATI Display Driver 7.983.2-040513a-015558C-IBM
Access IBM 4.0
Access IBM Tools 4.0
Adobe Reader 6.0.1 006.000.001
Agere Systems AC'97 Modem 2.1.31
```

Figure 10: The PsInfo *Command*

Cat (Linux)

The native *cat* command allows you to create a system profile by displaying file contents of the version, uptime, meminfo, filesystems, and cpuinfo files located in the proc file.

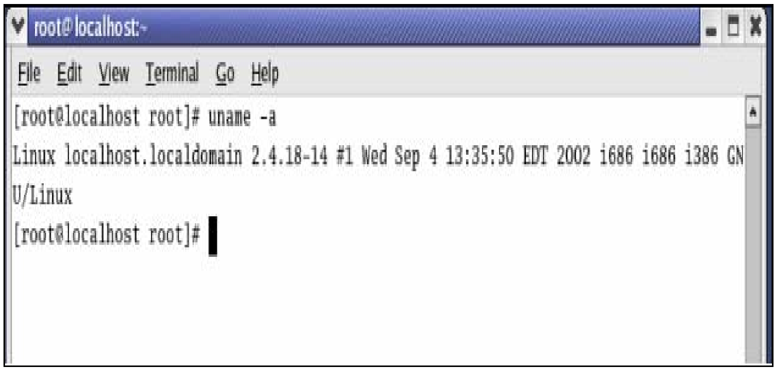


```
[root@localhost proc]# cat version uptime meminfo filesystems cpuinfo
Linux version 2.4.18-14 (bhcompile@stripples.devel.redhat.com) (gcc version 3.2 200209
03 (Red Hat Linux 8.0 3.2-7)) #1 Wed Sep 4 13:35:50 EDT 2002
3586.00 46.3396
        total:    used:    free:  shared: buffers:  cached:
Mem:  307015680 139620352 167395328        0 32858112 58363904
Swap: 534634496        0 534634496
MemTotal:       299820 kB
MemFree:        163472 kB
MemShared:           0 kB
Buffers:         32088 kB
```

Figure 11: The cat *Command*

Uname (Linux)

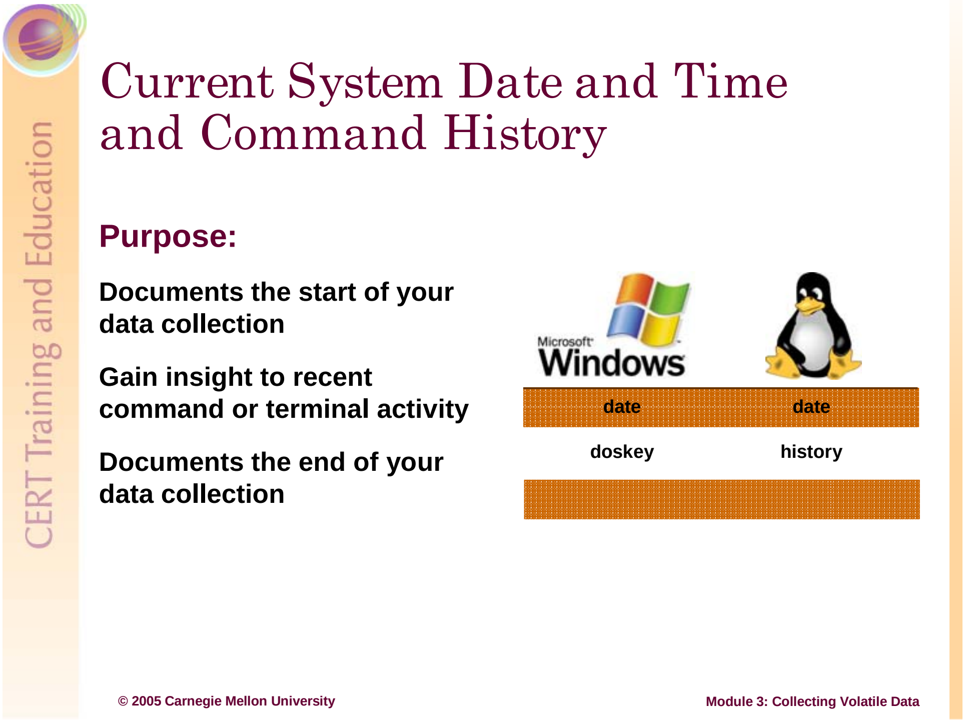
The native *uname* command allows you to create a system profile that includes machine name, machine's network node hostname, the type of processor, the OS release, and the OS kernel version.



```
[root@localhost root]# uname -a
Linux localhost.localdomain 2.4.18-14 #1 Wed Sep 4 13:35:50 EDT 2002 i686 i686 i386 GN
U/Linux
[root@localhost root]#
```

Figure 12: The uname *Command*

If you did not know the kernel version on the suspicious computer, this information is particularly important. An old kernel that hasn't been patched or recompiled may have vulnerabilities that are well documented by the vendor.



CERT Training and Education

Current System Date and Time and Command History

Purpose:

Documents the start of your data collection

Gain insight to recent command or terminal activity

Documents the end of your data collection

Microsoft Windows

date	**date**
doskey	**history**

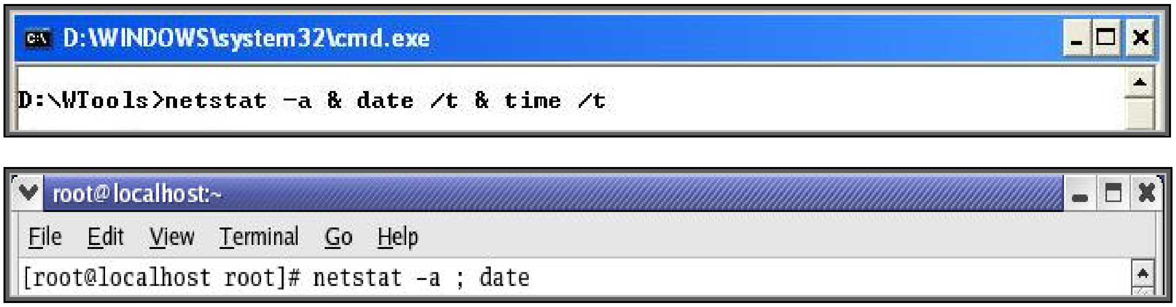
© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.9.1.2 Current System Date and Time and Command History

The system date, time, and command history are important throughout the forensic collection process. Initially, the first responder uses the current date and time to document when he discovered the suspicious computer and started collecting volatile data. During data collection, he uses the current date and time to document when each forensic tool or command was executed. These date and timestamps are critical to admissible data and easy to obtain from native OS commands and free open source tools.

Figure 13 shows how the native *date* and *time* commands can be used with a native forensic command like *netstat* to document at what time and date you executed the *netstat* command.



D:\WINDOWS\system32\cmd.exe

```
D:\WTools>netstat -a & date /t & time /t
```

root@localhost:~

```
[root@localhost root]# netstat -a ; date
```

Figure 13: date *and* time *Commands Used with* netstat

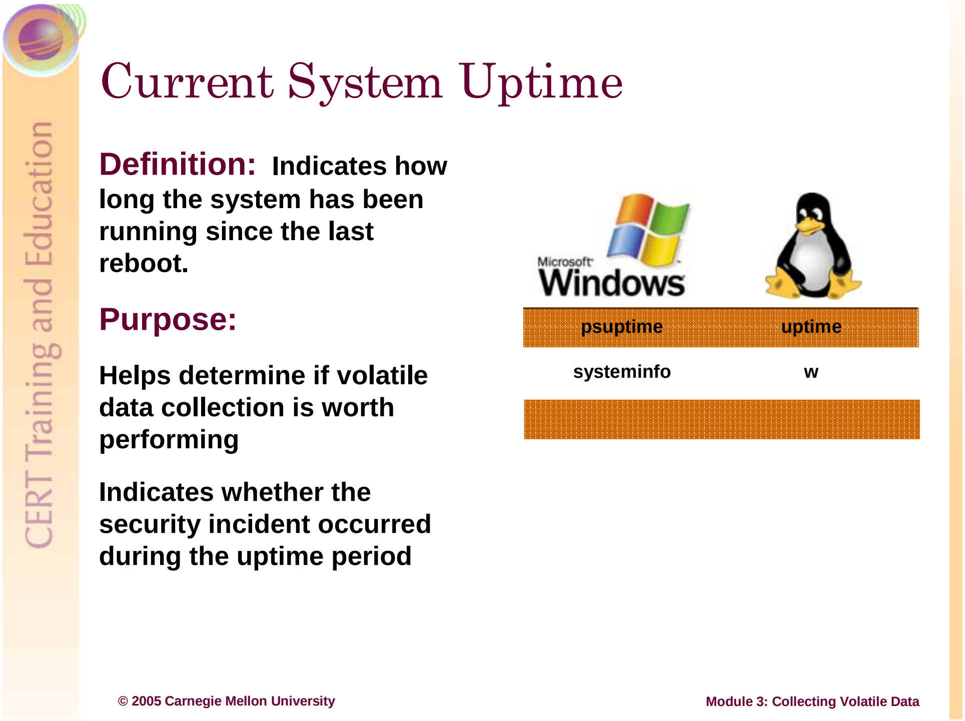
In both Linux and Windows the command history reveals recent activities of the suspicious computer's user(s). It includes a list of recently executed commands performed by a local or remote user within an established terminal or command shell. If the suspicious computer has more than one user account, collect the command history for each one. But be advised, Windows history log is virtual while Linux's is persistent.

A collected command history also functions as an audit trail of first responder actions. After you are done collecting data from the suspicious computer, use a native OS command or freeware utility to collect the command history.

As stated earlier, you lose the command history if the open Windows command shell is closed. The same is not true for Linux systems: you can still recover a command history if a terminal is closed.

As you collect a suspicious system's current date, time, and command history do the following:

- Determine if there is any discrepancy between the collected time and date and the actual time and date within your time zone.
- For the current command history, determine if administrative commands have been abused to
  - add user accounts
  - change NIC configuration
  - install unauthorized software applications
  - disable security logging, firewalls, or anti-virus software



CERT Training and Education

Current System Uptime

Definition: **Indicates how long the system has been running since the last reboot.**

Purpose:

Helps determine if volatile data collection is worth performing

Indicates whether the security incident occurred during the uptime period

psuptime	uptime
systeminfo	w

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

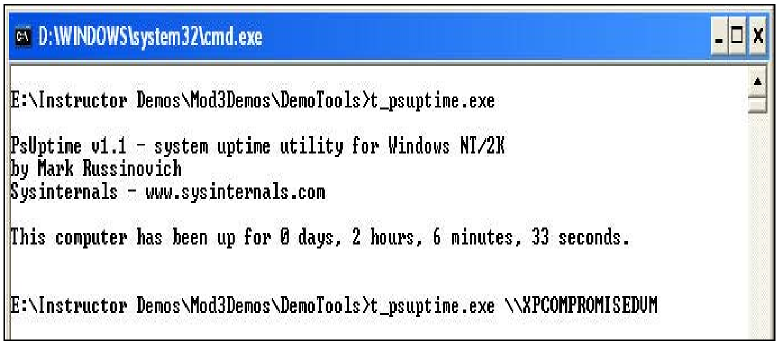
3.9.1.3 Current System Uptime

System uptime is the period of time that the system has been running since its last reboot. It can help the first responder determine if a volatile data collection is worth performing.

For example, if uptime is only a few minutes, the system has been rebooted recently and it may not be worthwhile to collect volatile data. And it may not be worthwhile if the security incident occurred before the beginning of the uptime period.

PsUptime **(Windows)**

SysInternals' *PsUptime* tells you how long a Windows system has been up. It uses the Performance Data Helper library to easily read the System Up Time performance counter on the local or remote computer.



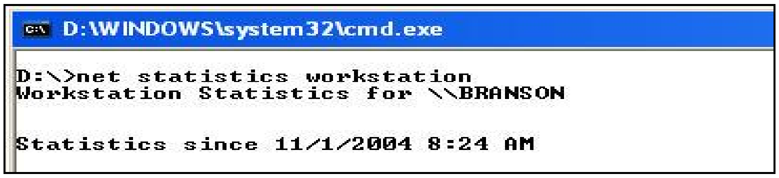
```
E:\Instructor Demos\Mod3Demos\DemoTools>t_psuptime.exe

PsUptime v1.1 - system uptime utility for Windows NT/2K
by Mark Russinovich
Sysinternals - www.sysinternals.com

This computer has been up for 0 days, 2 hours, 6 minutes, 33 seconds.


E:\Instructor Demos\Mod3Demos\DemoTools>t_psuptime.exe \\XPCOMPROMISEDUM
```

Figure 14: The PsUptime *Command*

To gather information on a remote computer, simply specify the computer's name with the command: psuptime.exe \\<computer name>)**.**

***Net Statistics* (Windows)**

The native *net statistics* command also allows you to collect uptime information (i.e., net statistics workstation or server).



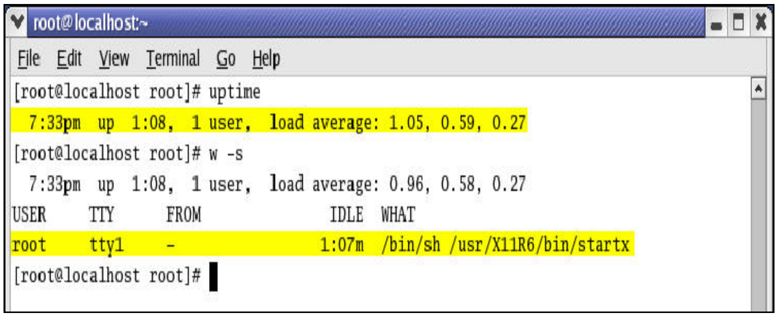
```
D:\>net statistics workstation
Workstation Statistics for \\BRANSON

Statistics since 11/1/2004 8:24 AM
```

Figure 15: The net statistics *Command*

***Uptime* and *W* (Linux)**

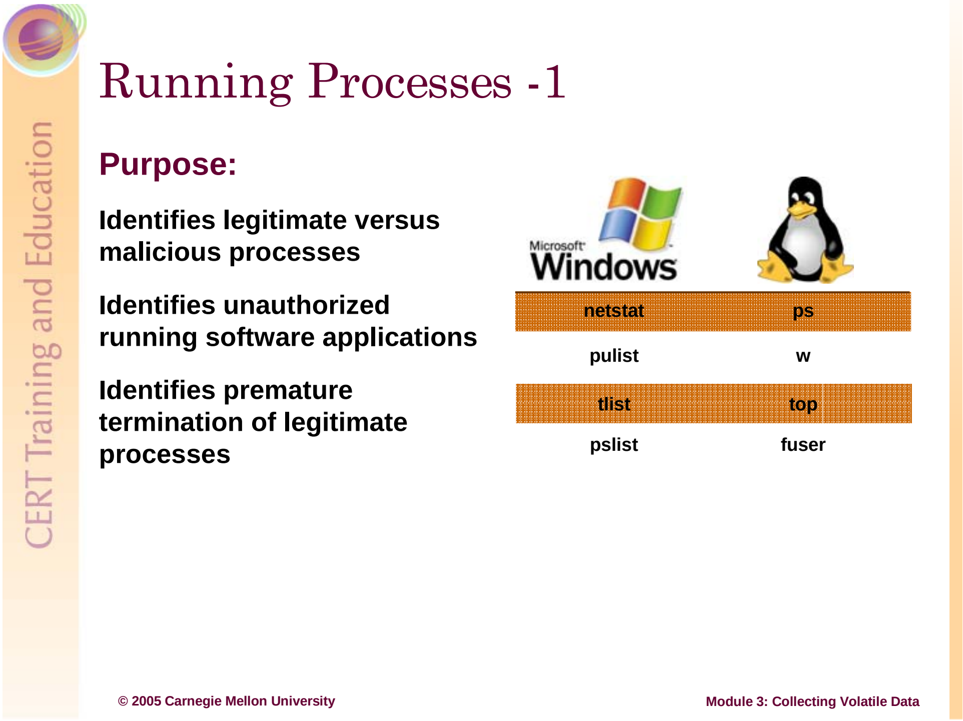
The native *uptime* and *w* commands allow you to collect uptime information.

In Figure 16, the machine has been up for approximately one hour and eight minutes from an initial logon at 7:33 p.m.



```
[root@localhost root]# uptime
  7:33pm  up  1:08,  1 user,  load average: 1.05, 0.59, 0.27
[root@localhost root]# w -s
  7:33pm  up  1:08,  1 user,  load average: 0.96, 0.58, 0.27
USER     TTY      FROM              IDLE  WHAT
root     tty1     -                1:07m  /bin/sh /usr/X11R6/bin/startx
[root@localhost root]#
```

Figure 16: The uptime *and* w *Commands*



CERT Training and Education

Running Processes -1

Purpose:

Identifies legitimate versus malicious processes

Identifies unauthorized running software applications

Identifies premature termination of legitimate processes

Microsoft Windows

netstat	**ps**
pulist	**w**
tlist	**top**
pslist	**fuser**

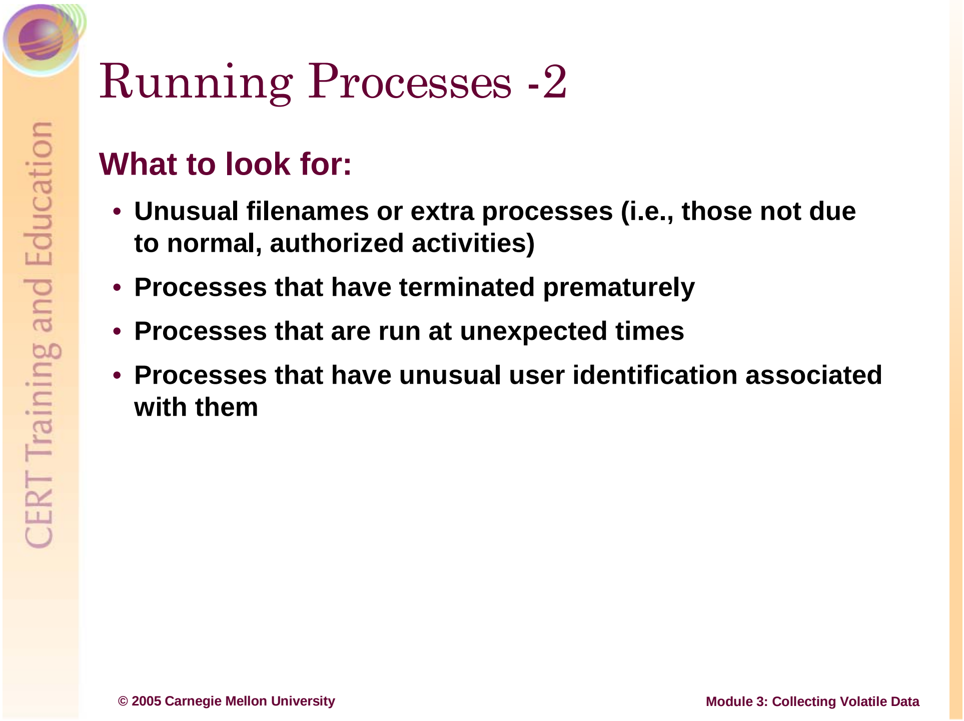
Module 3: Collecting Volatile Data

3.9.1.4 Running Processes

Examine the list of currently running processes to identify unnecessary or possible rogue (malicious) processes.

Rogue processes can be created and started independently and are often named for legitimate processes, which can make them extremely difficult to identify. To address this problem, first responders should compare each running process' PID (process identifier) or process filename to a list of legitimate system and application processes.

Uniblue [Uniblue 04] provides a list of legitimate processes for Windows systems on their Web site.



CERT Training and Education

Running Processes -2

What to look for:

- Unusual filenames or extra processes (i.e., those not due to normal, authorized activities)
- Processes that have terminated prematurely
- Processes that are run at unexpected times
- Processes that have unusual user identification associated with them

© 2005 Carnegie Mellon University

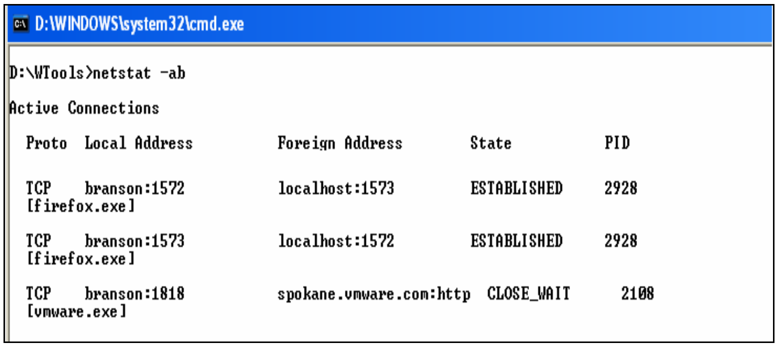
Module 3: Collecting Volatile Data

Harlan Carvey [Carvey 04] recommends documenting this set of characteristics for each running process:

- the process' executable image
- the command line used to initiate the process
- how long the process has been running
- the security context that it runs in
- modules or libraries (DLLs) it accesses
- memory that the process consumes

Currently, no single utility comprehensively assesses running processes. Therefore, the following examples show you how to use a combination of commands and utilities to assess process PID (2928).

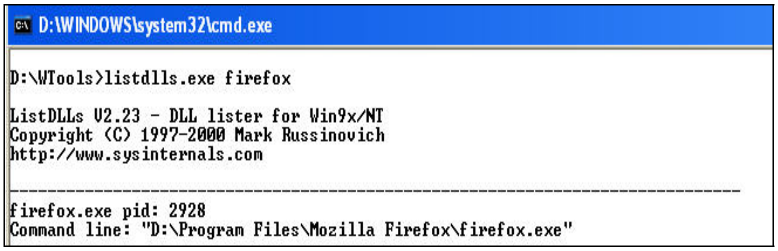
1. Use *netstat –ab* output to determine that the executable file for process PID (2928) is firefox.exe.



```
D:\WTools>netstat -ab

Active Connections

  Proto  Local Address          Foreign Address        State           PID

  TCP    branson:1572           localhost:1573         ESTABLISHED     2928
  [firefox.exe]

  TCP    branson:1573           localhost:1572         ESTABLISHED     2928
  [firefox.exe]

  TCP    branson:1818           spokane.vmware.com:http  CLOSE_WAIT      2108
  [vmware.exe]
```

Figure 17: Using netstat –ab *to Determine Process Executable Image*

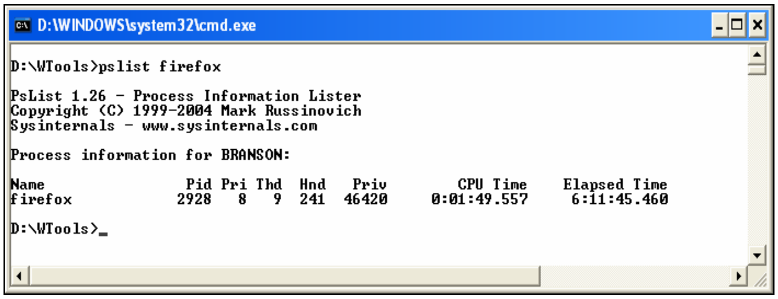
2. Use Sysinternals' *ListDLLs* to determine the command line used to execute the process.



```
D:\WTools>listdlls.exe firefox

ListDLLs V2.23 - DLL lister for Win9x/NT
Copyright (C) 1997-2000 Mark Russinovich
http://www.sysinternals.com

------------------------------------------------------------------------------
firefox.exe pid: 2928
Command line: "D:\Program Files\Mozilla Firefox\firefox.exe"
```

Figure 18: Using ListDLLs *to Determine Command Line*

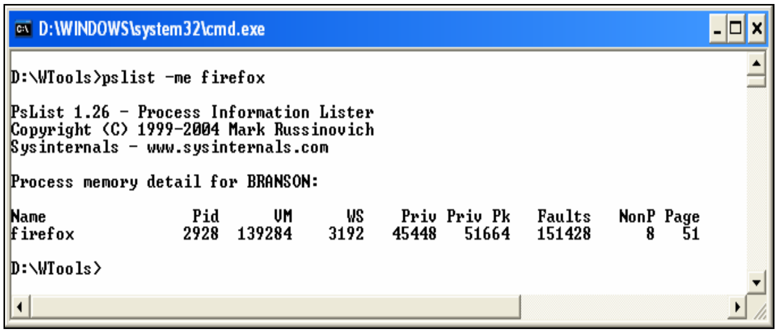
3. Use Sysinternals' *PsList* to discover how long firefox.exe (the 2928 process) has been running.



```
D:\WTools>pslist firefox

PsList 1.26 - Process Information Lister
Copyright (C) 1999-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

Process information for BRANSON:

Name                Pid Pri Thd  Hnd   Priv        CPU Time    Elapsed Time
firefox            2928   8   9  241  46420    0:01:49.557     6:11:45.460

D:\WTools>_
```

Figure 19: Using PsList *to Determine How Long a Process Has Been Running*

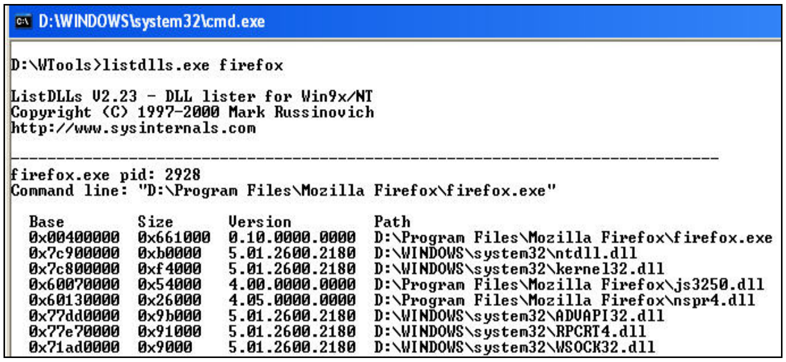
4. Use Sysinternals’ *PsList* to discover how much virtual memory the firefox.exe process is currently consuming.



```
D:\WTools>pslist -me firefox

PsList 1.26 - Process Information Lister
Copyright (C) 1999-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

Process memory detail for BRANSON:

Name                Pid      VM      WS    Priv  Priv Pk   Faults   NonP Page
firefox            2928  139284    3192   45448    51664   151428      8   51

D:\WTools>
```

Figure 20: Using PsList *to Determine How Much Virtual Memory a Process Is Using*

5. Use *ListDLLs* to discover the currently loaded DLLs for the firefox.exe process.



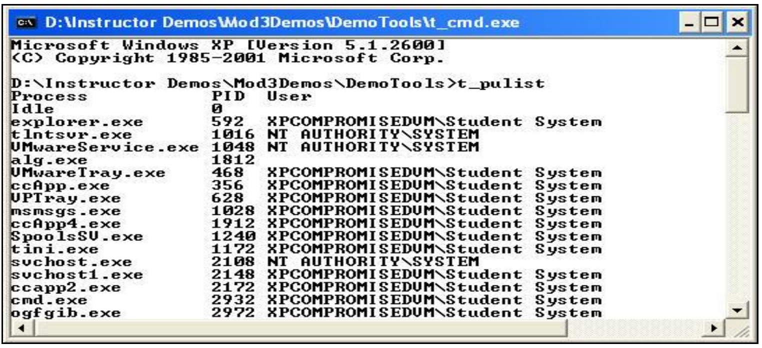
```
D:\WTools>listdlls.exe firefox

ListDLLs V2.23 - DLL lister for Win9x/NT
Copyright (C) 1997-2000 Mark Russinovich
http://www.sysinternals.com

------------------------------------------------------------------------------
firefox.exe pid: 2928
Command line: "D:\Program Files\Mozilla Firefox\firefox.exe"

  Base        Size      Version         Path
  0x00400000  0x661000  0.10.0000.0000  D:\Program Files\Mozilla Firefox\firefox.exe
  0x7c900000  0xb0000   5.01.2600.2180  D:\WINDOWS\system32\ntdll.dll
  0x7c800000  0xf4000   5.01.2600.2180  D:\WINDOWS\system32\kernel32.dll
  0x60070000  0x54000   4.00.0000.0000  D:\Program Files\Mozilla Firefox\js3250.dll
  0x60130000  0x26000   4.05.0000.0000  D:\Program Files\Mozilla Firefox\nspr4.dll
  0x77dd0000  0x9b000   5.01.2600.2180  D:\WINDOWS\system32\ADVAPI32.dll
  0x77e70000  0x91000   5.01.2600.2180  D:\WINDOWS\system32\RPCRT4.dll
  0x71ad0000  0x9000    5.01.2600.2180  D:\WINDOWS\system32\WSOCK32.dll
```

Figure 21: Using ListDLLs *to Discover the Currently Loaded DLLs for a Process*

Pulist (Windows)

Pulist output can be examined for unexpected process filenames and unusual user identifications. In Figure 22, the following four processes are well-known rogues: tini.exe, klogger.exe, svchost1.exe, and qfskrtwj.exe

.

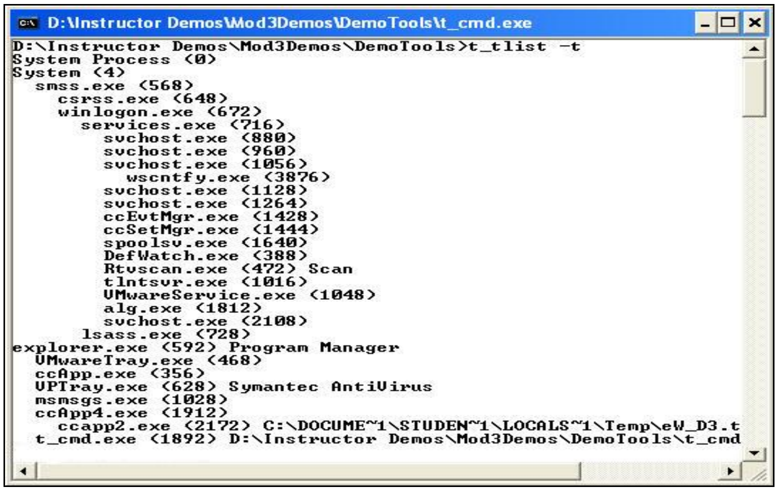


```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Instructor Demos\Mod3Demos\DemoTools>t_pulist
Process           PID  User
Idle              0
explorer.exe      592  XPCOMPROMISEDVM\Student System
tlntsvr.exe       1016 NT AUTHORITY\SYSTEM
VMwareService.exe 1048 NT AUTHORITY\SYSTEM
alg.exe           1812
VMwareTray.exe    468  XPCOMPROMISEDVM\Student System
ccApp.exe         356  XPCOMPROMISEDVM\Student System
VPTray.exe        628  XPCOMPROMISEDVM\Student System
msmsgs.exe        1028 XPCOMPROMISEDVM\Student System
ccApp4.exe        1912 XPCOMPROMISEDVM\Student System
SpoolsSV.exe      1240 XPCOMPROMISEDVM\Student System
tini.exe          1172 XPCOMPROMISEDVM\Student System
svchost.exe       2108 NT AUTHORITY\SYSTEM
svchost1.exe      2148 XPCOMPROMISEDVM\Student System
ccapp2.exe        2172 XPCOMPROMISEDVM\Student System
cmd.exe           2932 XPCOMPROMISEDVM\Student System
ogfgib.exe        2972 XPCOMPROMISEDVM\Student System
```

Figure 22: Pulist *Output*

Tlist (Windows)

Windows NT Server Resource Kit includes *tlist.exe*, which lists active tasks and processes.

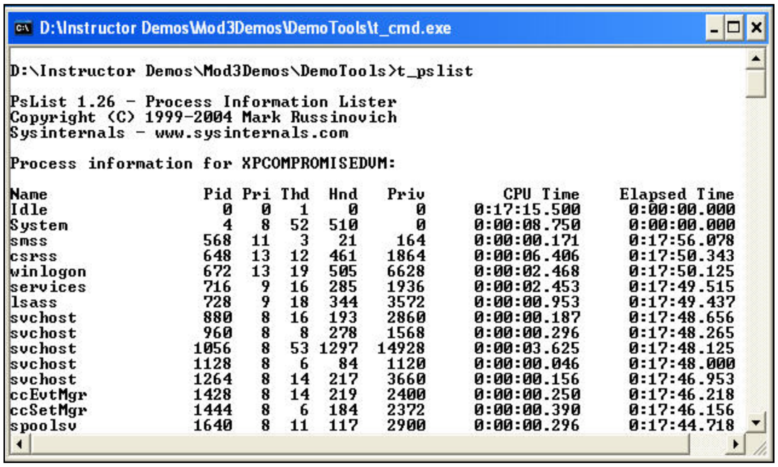


```
D:\Instructor Demos\Mod3Demos\DemoTools>t_tlist -t
System Process (0)
System (4)
  smss.exe (568)
    csrss.exe (648)
    winlogon.exe (672)
      services.exe (716)
        svchost.exe (880)
        svchost.exe (960)
        svchost.exe (1056)
          wscntfy.exe (3876)
        svchost.exe (1128)
        svchost.exe (1264)
        ccEvtMgr.exe (1428)
        ccSetMgr.exe (1444)
        spoolsv.exe (1640)
        DefWatch.exe (388)
        Rtvscan.exe (472) Scan
        tlntsvr.exe (1016)
        VMwareService.exe (1048)
        alg.exe (1812)
        svchost.exe (2108)
      lsass.exe (728)
explorer.exe (592) Program Manager
  VMwareTray.exe (468)
  ccApp.exe (356)
  VPTray.exe (628) Symantec AntiVirus
  msmsgs.exe (1028)
  ccApp4.exe (1912)
    ccapp2.exe (2172) C:\DOCUME~1\STUDEN~1\LOCALS~1\Temp\eW_D3.t
  t_cmd.exe (1892) D:\Instructor Demos\Mod3Demos\DemoTools\t_cmd
```

Figure 23: tlist.exe

PsList (Windows)

Sysinternals’ *PsList* combines outputs from *pmon* and *pstat* to show process CPU and memory information (i.e., thread statistics). *PsList*, unlike *pmon* and *pstat*, allows you to view process and thread statistics on a remote computer.



```
D:\Instructor Demos\Mod3Demos\DemoTools>t_pslist

PsList 1.26 - Process Information Lister
Copyright (C) 1999-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

Process information for XPCOMPROMISEDVM:

Name                Pid Pri Thd  Hnd   Priv        CPU Time    Elapsed Time
Idle                  0   0   1    0      0     0:17:15.500     0:00:00.000
System                4   8  52  510      0     0:00:08.750     0:00:00.000
smss                568  11   3   21    164     0:00:00.171     0:17:56.078
csrss               648  13  12  461   1864     0:00:06.406     0:17:50.343
winlogon            672  13  19  505   6628     0:00:02.468     0:17:50.125
services            716   9  16  285   1936     0:00:02.453     0:17:49.515
lsass               728   9  18  344   3572     0:00:00.953     0:17:49.437
svchost             880   8  16  193   2860     0:00:00.187     0:17:48.656
svchost             960   8   8  278   1568     0:00:00.296     0:17:48.265
svchost            1056   8  53 1297  14928     0:00:03.625     0:17:48.125
svchost            1128   8   6   84   1120     0:00:00.046     0:17:48.000
svchost            1264   8  14  217   3660     0:00:00.156     0:17:46.953
ccEvtMgr           1428   8  14  219   2400     0:00:00.250     0:17:46.218
ccSetMgr           1444   8   6  184   2372     0:00:00.390     0:17:46.156
spoolsv            1640   8  11  117   2900     0:00:00.296     0:17:44.718
```

Figure 24: PsList

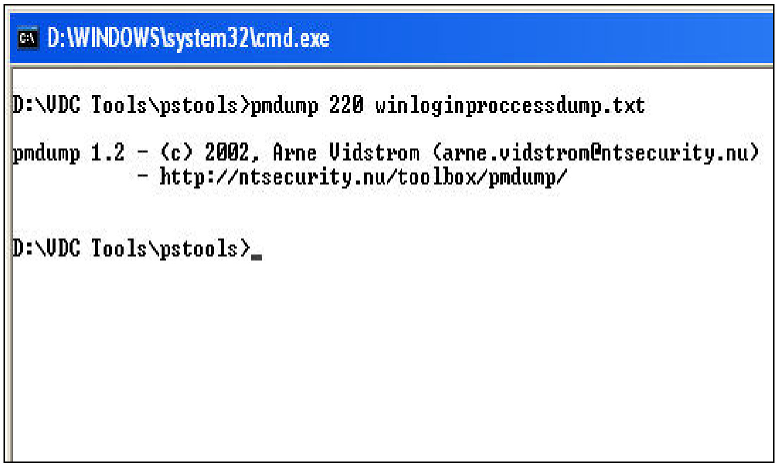
Table 2 shows definitions for the *PsList* utilities’ command output headings.

Table 2: PsList Output Headings

Pri:	Priority	WS:	Working set
Thd:	Number of threads	WSPk:	Working set peak
Hnd:	Number of handles	Priv:	Private memory
Mem:	Memory usage	Faults:	Page faults
VM:	Virtual memory	NonP:	Non-paged memory

Memory Dump

To learn more about a suspected rogue process, create a process memory dump using the *pmdump.exe* utility and then perform string searches on the file (e.g., use the Ctrl F command or the *strings* utility). Figure 25 shows the command string required to create a dump of the running process called *winlogin.*



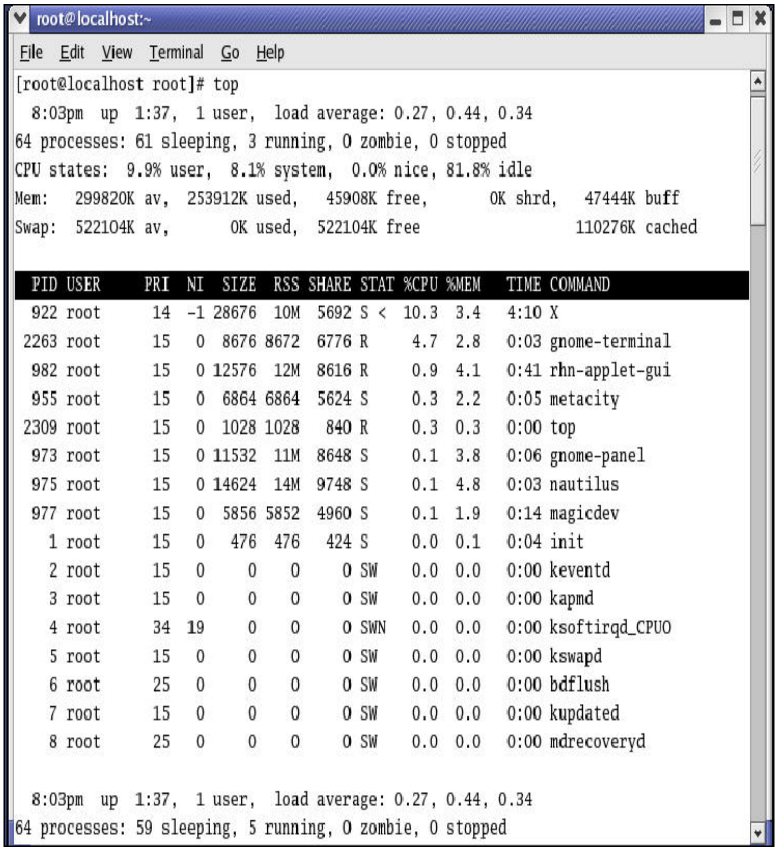
```
D:\VDC Tools\pstools>pmdump 220 winloginproccessdump.txt

pmdump 1.2 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
           - http://ntsecurity.nu/toolbox/pmdump/


D:\VDC Tools\pstools>_
```

Figure 25: A Process Memory Dump

A better alternative is to export the running process’ executable image to a forensic tool testbed for closer analysis.

Top (Linux)

The *top* command provides an ongoing look at processor activity in real time. It lists the most CPU-intensive tasks and includes an interactive interface for sorting processes by CPU usage, memory usage and runtime. Sort features can also be specified in the personal or system-wide configuration files.



```
[root@localhost root]# top
  8:03pm  up  1:37,  1 user,  load average: 0.27, 0.44, 0.34
64 processes: 61 sleeping, 3 running, 0 zombie, 0 stopped
CPU states:  9.9% user,  8.1% system,  0.0% nice, 81.8% idle
Mem:   299820K av,  253912K used,   45908K free,       0K shrd,   47444K buff
Swap:  522104K av,       0K used,  522104K free                  110276K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
  922 root      14  -1 28676  10M  5692 S <  10.3  3.4   4:10 X
 2263 root      15   0  8676 8672  6776 R     4.7  2.8   0:03 gnome-terminal
  982 root      15   0 12576  12M  8616 R     0.9  4.1   0:41 rhn-applet-gui
  955 root      15   0  6864 6864  5624 S     0.3  2.2   0:05 metacity
 2309 root      15   0  1028 1028   840 R     0.3  0.3   0:00 top
  973 root      15   0 11532  11M  8648 S     0.1  3.8   0:06 gnome-panel
  975 root      15   0 14624  14M  9748 S     0.1  4.8   0:03 nautilus
  977 root      15   0  5856 5852  4960 S     0.1  1.9   0:14 magicdev
    1 root      15   0   476  476   424 S     0.0  0.1   0:04 init
    2 root      15   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root      15   0     0    0     0 SW    0.0  0.0   0:00 kapmd
    4 root      34  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    5 root      15   0     0    0     0 SW    0.0  0.0   0:00 kswapd
    6 root      25   0     0    0     0 SW    0.0  0.0   0:00 bdflush
    7 root      15   0     0    0     0 SW    0.0  0.0   0:00 kupdated
    8 root      25   0     0    0     0 SW    0.0  0.0   0:00 mdrecoveryd

  8:03pm  up  1:37,  1 user,  load average: 0.27, 0.44, 0.34
64 processes: 59 sleeping, 5 running, 0 zombie, 0 stopped
```

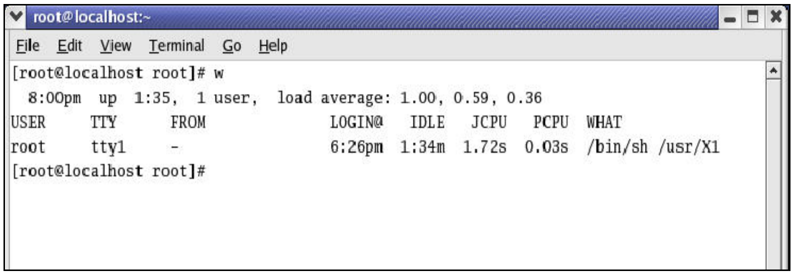
Figure 26: The top *Command*

Be aware that *top* can be configured to refresh at regular intervals so that results created immediately after each other may be different.

While *top* is running, press the appropriate key to update speed (s), hide idle processes (i), or kill a processes (k).

W (Linux)

The *w* command displays the current processes for each shell of each user.



```
[root@localhost root]# w
  8:00pm  up  1:35,  1 user,  load average: 1.00, 0.59, 0.36
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
root     tty1     -                 6:26pm  1:34m  1.72s  0.03s  /bin/sh /usr/X1
[root@localhost root]#
```

Figure 27: The w *Command*

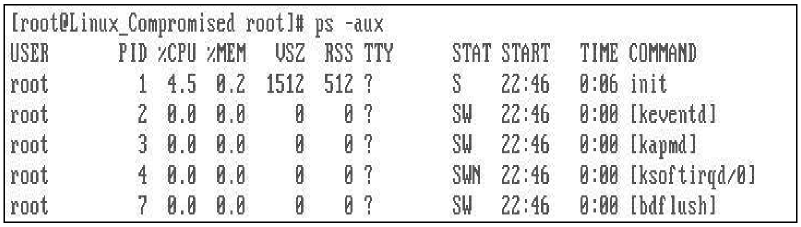
Output includes the following fields [Mandia 01]:

Table 3: w Command Output Fields

Field	**Description**
TTY	Control terminal assigned to the users session. A *ttyn* (where n is zero or a positive integer) signifies a logon at the console (a user logging on to the system from the local console or keyboard). A *ptsn* or *ttypn* may signify a connection over the network.
FROM	Fully qualified domain name or numerical IP address of the remote host. A hyphen (-) in this field corresponds to a local logon at console.
LOGIN@	Local starting time of the connection
IDLE	Length of time since the last process was run
JCPU	Time used by all processes attached to the *tty* or *pts*
PCPU	Processor time used by the current process in the WHAT column
WHAT	Process that the user is currently running

Ps (Linux)

The *ps* command, along with the *netstat* command, is one of the most powerful Linux administrative management commands.



```
[root@Linux_Compromised root]# ps -aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  4.5  0.2  1512  512 ?        S    22:46   0:06 init
root         2  0.0  0.0     0    0 ?        SW   22:46   0:00 [keventd]
root         3  0.0  0.0     0    0 ?        SW   22:46   0:00 [kapmd]
root         4  0.0  0.0     0    0 ?        SWN  22:46   0:00 [ksoftirqd/0]
root         7  0.0  0.0     0    0 ?        SW   22:46   0:00 [bdflush]
```

Figure 28: The ps *Command*

Figure 28 displays information about the root's currently running processes. A subset of the *ps* command's 80 options that are most useful to first responders are listed in Table 4 [Barrett 04].

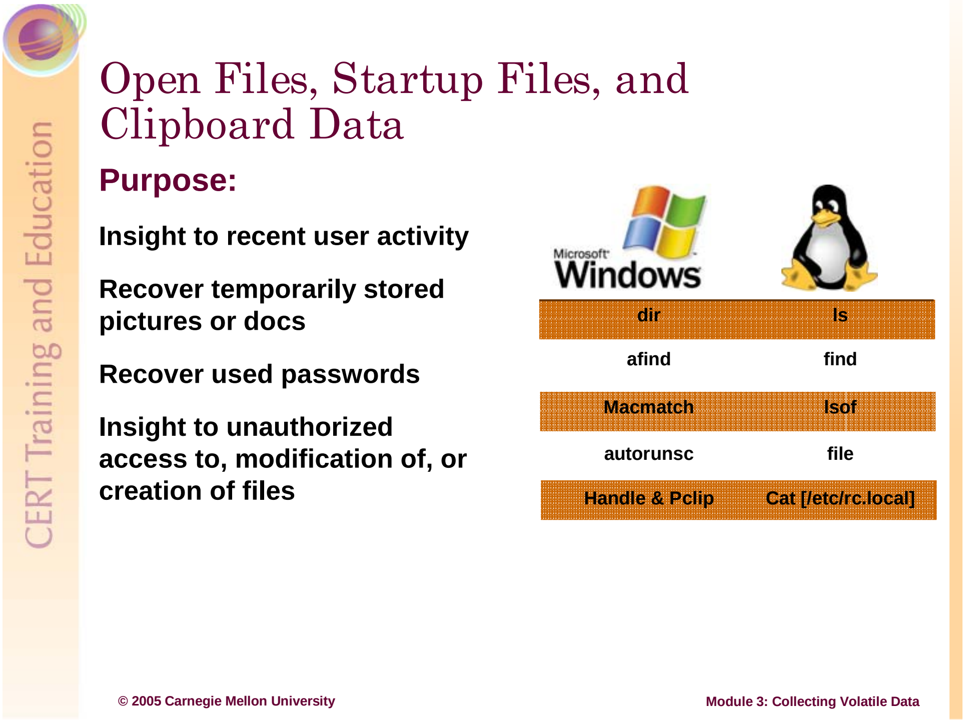
Table 4: A Subset of ps *Options*

Option	**Description**
$ ps –ux	View current processes
$ ps –U *user*	View other system users running processes
$ ps –C *program_name*	View all occurrences of a program
$ ps –p4, 8, 2203	View selected processes 4, 8, 2203
$ ps –efww	View all processes with full command lines

Table 5 describes some output headings for *ps* and *top* output [Nemeth 01].

Table 5: Output Headings for ps *and* top

Field	**Description**
USER	Username of the processes owner
PID	Process ID
%CPU	Percentage of the CPU this process is using
%MEM	Percentage of real memory this process is using
VSZ	Virtual size of the process, in kilobytes
RSS	
TT	
STAT	**Current Process Status** R= Runnable D = In disk wait I = Sleeping (< 20 sec) S = Sleeping (> 20 sec) T = Stopped Z = Zombie **Additional Flags** L = Some pages are locked in core (for rawio) S = Process is a session leader (head of control terminal) W = Process is swapped out + = Process is in the foreground of its control terminal
START	Time the process was started
TIME	CPU time the process has consumed
COMMAND	Command name and arguments



CERT Training and Education

Open Files, Startup Files, and Clipboard Data

Purpose:

Insight to recent user activity

Recover temporarily stored pictures or docs

Recover used passwords

Insight to unauthorized access to, modification of, or creation of files

Microsoft Windows	
dir	ls
afind	find
Macmatch	lsof
autorunsc	file
Handle & Pclip	Cat [/etc/rc.local]

Module 3: Collecting Volatile Data

3.9.1.5 Open Files, Startup Files, and Clipboard Data

When malware such as a Trojan or Rootkit infects a system, files are created that allow the malware to start (when the system is rebooted) or continue to run (when the user logs out). For example, such files are generally created with the startup folders for Windows computers or within the rc.local file for Linux.

Therefore, collect and document the following data:

- open or recently created documents
- startup files and programs
- Dates and Times associated with MAC times for critical folders and files like Windows System32 folder and Linux's proc file

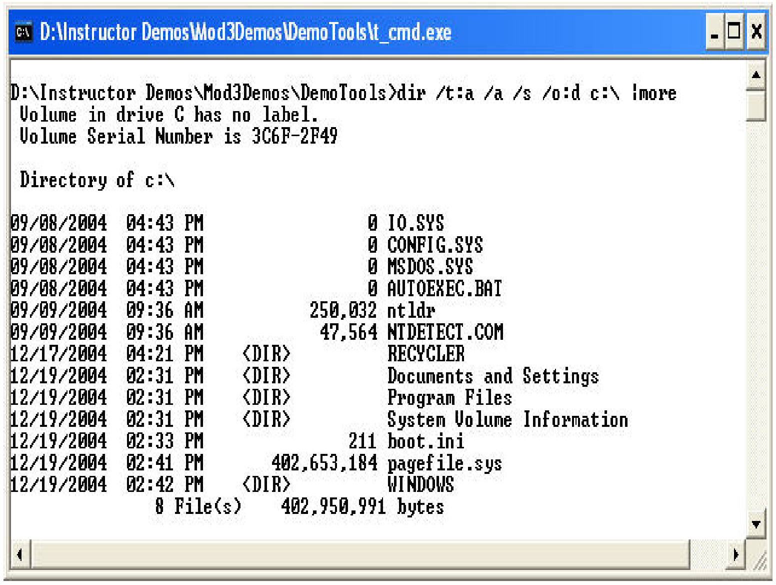
Examine this data as follows:

- Review open files for sensitive data or information that may be relevant to the security incident.
- Look for valuable data like passwords, images, and pieces of documents on the clipboard.
- Search for unusual MAC times on critical folders and startup files.

***Dir* (Windows)**

The native *dir* command lists the last access times on files and folders.

Figure 29 shows how to collect the last access times on all files and folders within the c:\ drive. With options, you can specify certain folders (e.g., System32).



```
D:\Instructor Demos\Mod3Demos\DemoTools>dir /t:a /a /s /o:d c:\ |more
 Volume in drive C has no label.
 Volume Serial Number is 3C6F-2F49

 Directory of c:\

09/08/2004  04:43 PM                 0 IO.SYS
09/08/2004  04:43 PM                 0 CONFIG.SYS
09/08/2004  04:43 PM                 0 MSDOS.SYS
09/08/2004  04:43 PM                 0 AUTOEXEC.BAT
09/09/2004  09:36 AM           250,032 ntldr
09/09/2004  09:36 AM            47,564 NTDETECT.COM
12/17/2004  04:21 PM    <DIR>          RECYCLER
12/19/2004  02:31 PM    <DIR>          Documents and Settings
12/19/2004  02:31 PM    <DIR>          Program Files
12/19/2004  02:31 PM    <DIR>          System Volume Information
12/19/2004  02:33 PM               211 boot.ini
12/19/2004  02:41 PM       402,653,184 pagefile.sys
12/19/2004  02:42 PM    <DIR>          WINDOWS
               8 File(s)    402,950,991 bytes
```

Figure 29: The dir *Command*

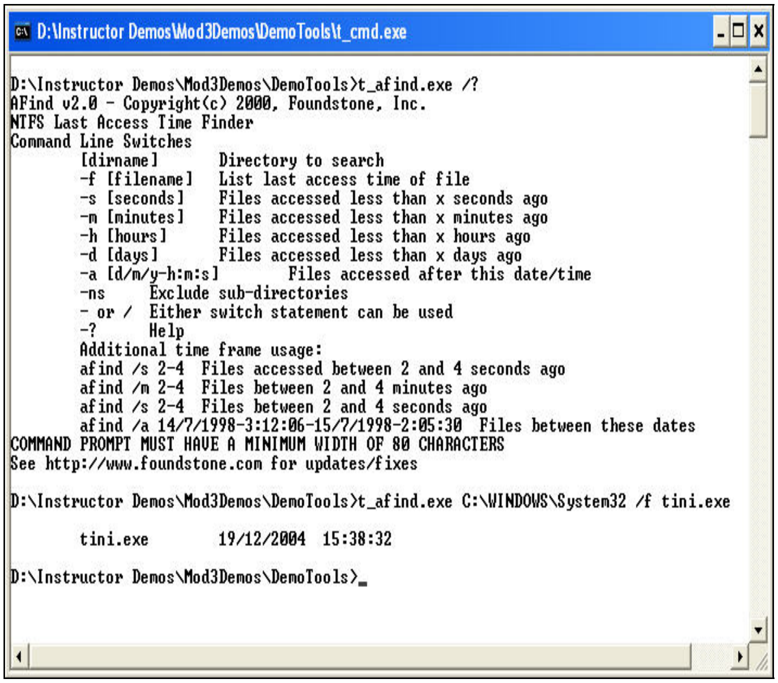
The native *dir* command includes the following options:

Table 6: dir *Command Options*

Option	**Description**
/t: a	Displays the last access field to be displayed and used for sorting
/a	Displays all files….Plus any hidden files
/s	Displays files in specified directory and all subdirectories
/o:d	Displays files in sorted order by date

Afind (Windows)

Figure 30 shows how to search for a file's access time within a specified location.



```
D:\Instructor Demos\Mod3Demos\DemoTools>t_afind.exe /?
AFind v2.0 - Copyright(c) 2000, Foundstone, Inc.
NTFS Last Access Time Finder
Command Line Switches
        [dirname]       Directory to search
        -f [filename]   List last access time of file
        -s [seconds]    Files accessed less than x seconds ago
        -m [minutes]    Files accessed less than x minutes ago
        -h [hours]      Files accessed less than x hours ago
        -d [days]       Files accessed less than x days ago
        -a [d/m/y-h:m:s]        Files accessed after this date/time
        -ns     Exclude sub-directories
        - or /  Either switch statement can be used
        -?      Help
        Additional time frame usage:
        afind /s 2-4  Files accessed between 2 and 4 seconds ago
        afind /m 2-4  Files between 2 and 4 minutes ago
        afind /s 2-4  Files between 2 and 4 seconds ago
        afind /a 14/7/1998-3:12:06-15/7/1998-2:05:30  Files between these dates
COMMAND PROMPT MUST HAVE A MINIMUM WIDTH OF 80 CHARACTERS
See http://www.foundstone.com for updates/fixes

D:\Instructor Demos\Mod3Demos\DemoTools>t_afind.exe C:\WINDOWS\System32 /f tini.exe

        tini.exe        19/12/2004  15:38:32

D:\Instructor Demos\Mod3Demos\DemoTools>_
```

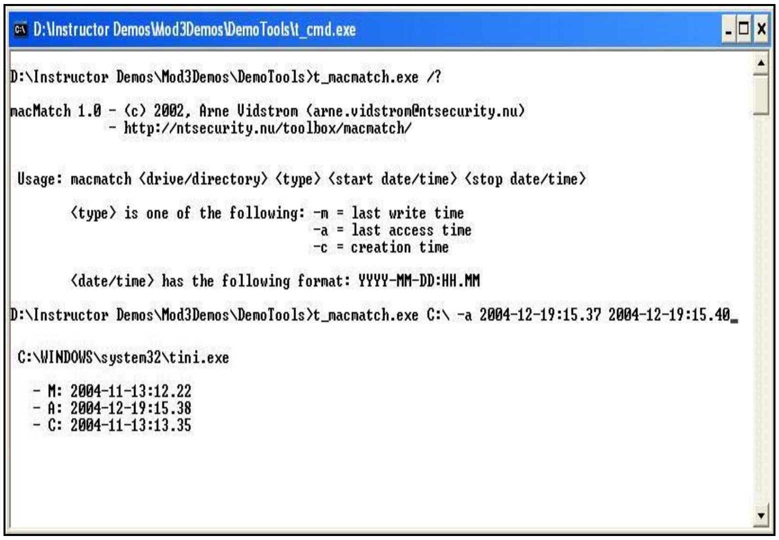
Figure 30: The afind *Command*

By specifying the general time period of the security incident, you can compare access times with logon information provided from *ntlast* and begin to determine user activity even if file logging had not been enabled.

MACMatch (Windows)

NTSecurity.nu-developed *MACMatch* allows you to search for files by their last write, last access, or creation time without changing any of these times [Vidstrom 04].

Usage is shown in Figure 31, which includes a search within a certain timeframe for recently accessed files such as the *tini.exe*.

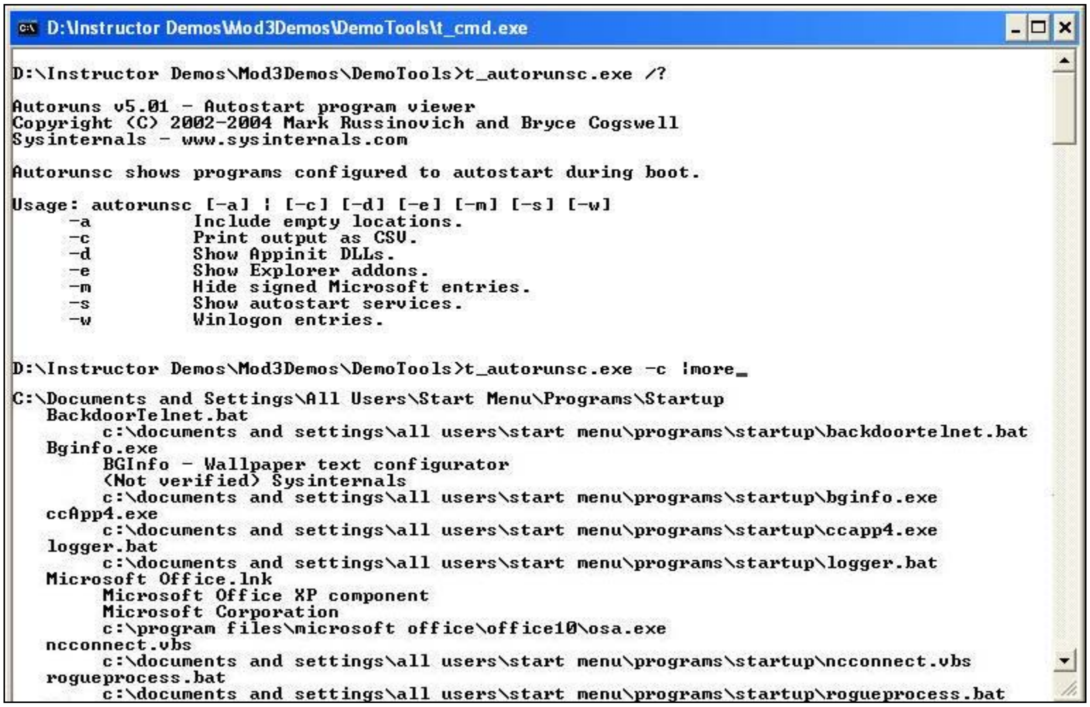


```
D:\Instructor Demos\Mod3Demos\DemoTools>t_macmatch.exe /?

macMatch 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
             - http://ntsecurity.nu/toolbox/macmatch/

 Usage: macmatch <drive/directory> <type> <start date/time> <stop date/time>

        <type> is one of the following: -m = last write time
                                        -a = last access time
                                        -c = creation time

        <date/time> has the following format: YYYY-MM-DD:HH.MM

D:\Instructor Demos\Mod3Demos\DemoTools>t_macmatch.exe C:\ -a 2004-12-19:15.37 2004-12-19:15.40_

 C:\WINDOWS\system32\tini.exe

   - M: 2004-11-13:12.22
   - A: 2004-12-19:15.38
   - C: 2004-11-13:13.35
```

Figure 31: MACMatch

Autorunsc (Windows)

Sysinternals' *Autoruns* and DiamondCS' *Autostart* are freeware utilities that list files in the startup areas. Both incorporate a GUI, which is a drawback: *Autorunsc* is *Autoruns* command-line version. All three products provide much more detail than Windows' *Msconfig* utility.



```
D:\Instructor Demos\Mod3Demos\DemoTools>t_autorunsc.exe /?

Autoruns v5.01 - Autostart program viewer
Copyright (C) 2002-2004 Mark Russinovich and Bryce Cogswell
Sysinternals - www.sysinternals.com

Autorunsc shows programs configured to autostart during boot.

Usage: autorunsc [-a] | [-c] [-d] [-e] [-m] [-s] [-w]
      -a          Include empty locations.
      -c          Print output as CSV.
      -d          Show Appinit DLLs.
      -e          Show Explorer addons.
      -m          Hide signed Microsoft entries.
      -s          Show autostart services.
      -w          Winlogon entries.


D:\Instructor Demos\Mod3Demos\DemoTools>t_autorunsc.exe -c |more_
C:\Documents and Settings\All Users\Start Menu\Programs\Startup
   BackdoorTelnet.bat
        c:\documents and settings\all users\start menu\programs\startup\backdoortelnet.bat
   Bginfo.exe
        BGInfo - Wallpaper text configurator
        (Not verified) Sysinternals
        c:\documents and settings\all users\start menu\programs\startup\bginfo.exe
   ccApp4.exe
        c:\documents and settings\all users\start menu\programs\startup\ccapp4.exe
   logger.bat
        c:\documents and settings\all users\start menu\programs\startup\logger.bat
   Microsoft Office.lnk
        Microsoft Office XP component
        Microsoft Corporation
        c:\program files\microsoft office\office10\osa.exe
   ncconnect.vbs
        c:\documents and settings\all users\start menu\programs\startup\ncconnect.vbs
   rogueprocess.bat
        c:\documents and settings\all users\start menu\programs\startup\rogueprocess.bat
```

Figure 32: Autorunsc

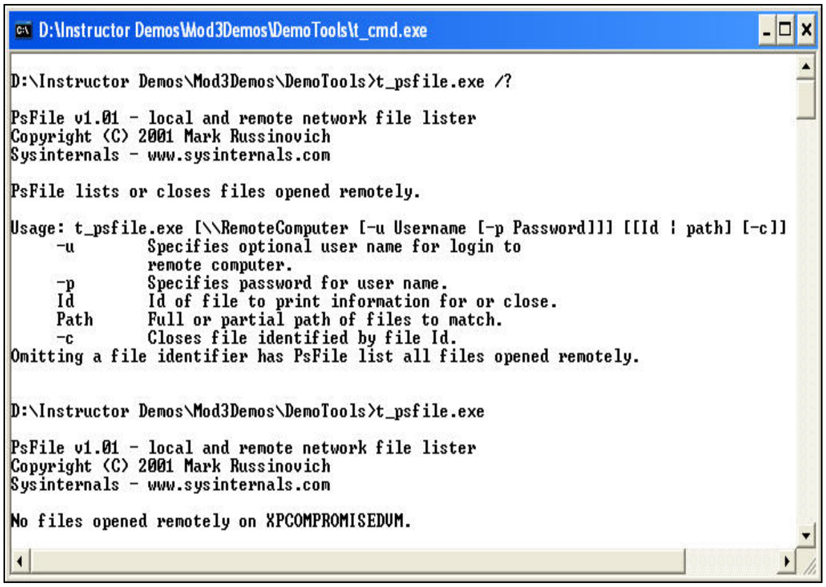
Autoruns has a comprehensive knowledge of auto-starting locations and shows what programs are configured to run during system bootup or login in the order Windows processes them. Programs include ones in your startup folder and Run, RunOnce, and other registry keys. You can configure Autoruns to show other locations, including Windows Explorer shell extensions, toolbars, browser helper objects, winlogon notifications, auto-start services, and much more.

> *Autoruns' "Hide Signed Microsoft Entries" option helps you to zoom in on third-party auto-starting images that have been added to your system. Also included in the download package is a command-line equivalent that can output in CSV format, Autorunsc [Russinovich 04b].*

PsFile and *Handle* (Windows)

Unsaved files or edits are lost after the system is shut down. Sysinternals' *PsFile* and *Handle* utilities allow you to display open files (executables and user-created), programs with open files, and the names of all program handles.

PsFile is a command-line utility that shows a list of files on a system that are opened remotely, and it also allows you to close opened files either by name or by a file identifier [Russinovich 01].

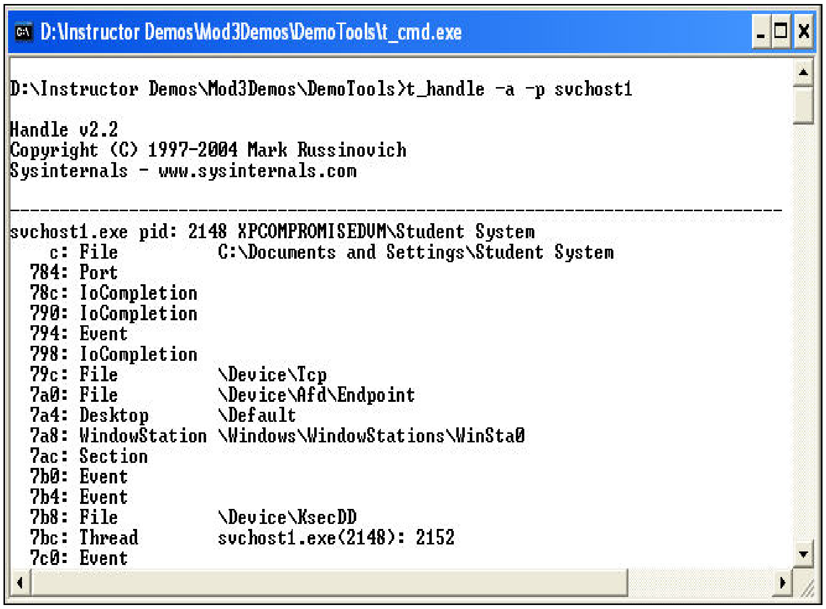


```
D:\Instructor Demos\Mod3Demos\DemoTools>t_psfile.exe /?

PsFile v1.01 - local and remote network file lister
Copyright (C) 2001 Mark Russinovich
Sysinternals - www.sysinternals.com

PsFile lists or closes files opened remotely.

Usage: t_psfile.exe [\\RemoteComputer [-u Username [-p Password]]] [[Id | path] [-c]]
     -u        Specifies optional user name for login to
               remote computer.
     -p        Specifies password for user name.
     Id        Id of file to print information for or close.
     Path      Full or partial path of files to match.
     -c        Closes file identified by file Id.
Omitting a file identifier has PsFile list all files opened remotely.


D:\Instructor Demos\Mod3Demos\DemoTools>t_psfile.exe

PsFile v1.01 - local and remote network file lister
Copyright (C) 2001 Mark Russinovich
Sysinternals - www.sysinternals.com

No files opened remotely on XPCOMPROMISEDVM.
```

Figure 33: PsFile

Handle is a utility that displays information about open handles for any process in the system. You can use it to see the programs that have a file open, or to see the object types and names of all the handles of a program [Russinovich 04c].

In Figure 34 we demonstrate how we can utilize the *handle.exe* utility to look at all open handles for the potential rogue process *svchost1*.



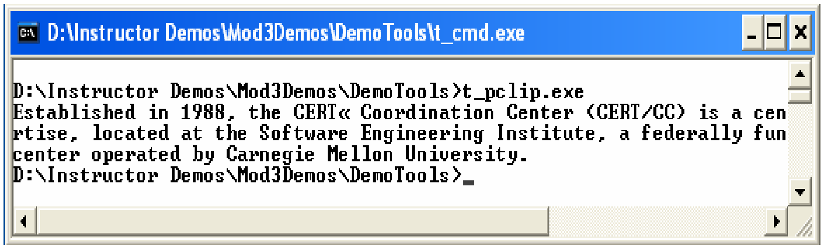
```
D:\Instructor Demos\Mod3Demos\DemoTools>t_handle -a -p svchost1

Handle v2.2
Copyright (C) 1997-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

------------------------------------------------------------------------------
svchost1.exe pid: 2148 XPCOMPROMISEDVM\Student System
    c: File          C:\Documents and Settings\Student System
  784: Port
  78c: IoCompletion
  790: IoCompletion
  794: Event
  798: IoCompletion
  79c: File          \Device\Tcp
  7a0: File          \Device\Afd\Endpoint
  7a4: Desktop       \Default
  7a8: WindowStation \Windows\WindowStations\WinSta0
  7ac: Section
  7b0: Event
  7b4: Event
  7b8: File          \Device\KsecDD
  7bc: Thread        svchost1.exe(2148): 2152
  7c0: Event
```

Figure 34: Handle

Pclip (Windows)

Two freeware utilities, GNU's *pclip* utility and Harlan Carvey's Perl script, allow you to dump the Windows clipboard contents: a document fragment, password, or image.

> *The clipboard is nothing more than an area of memory for temporarily holding information [Carvey 04].*



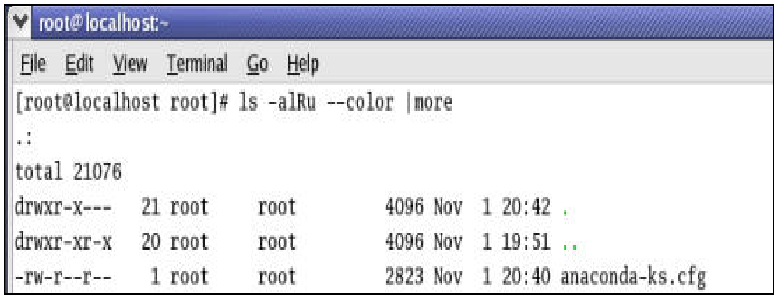
```
D:\Instructor Demos\Mod3Demos\DemoTools>t_pclip.exe
Established in 1988, the CERT« Coordination Center (CERT/CC) is a cen
rtise, located at the Software Engineering Institute, a federally fun
center operated by Carnegie Mellon University.
D:\Instructor Demos\Mod3Demos\DemoTools>_
```

Figure 35: Pclip

In Figure 35, *pclip* displays a copied text string.

Ls (Linux)

The native *ls* command, like the Windows *dir* command, allows you to list files and their file access times. In Figure 36, ls –alRu displays the last access times for files in the root folder.



```
[root@localhost root]# ls -alRu --color |more
.:
total 21076
drwxr-x---   21 root     root        4096 Nov  1 20:42 .
drwxr-xr-x   20 root     root        4096 Nov  1 19:51 ..
-rw-r--r--    1 root     root        2823 Nov  1 20:40 anaconda-ks.cfg
```

Figure 36: The ls *Command*

For Linux, three different permissions are tracked for every file and folder on the hard drive: read (r), write (w), and execute (x). To check file permissions for all the files in your current working directory, enter the following in a terminal window:

```
# ls –la
```

A sample output is as follows [CAE 04]:

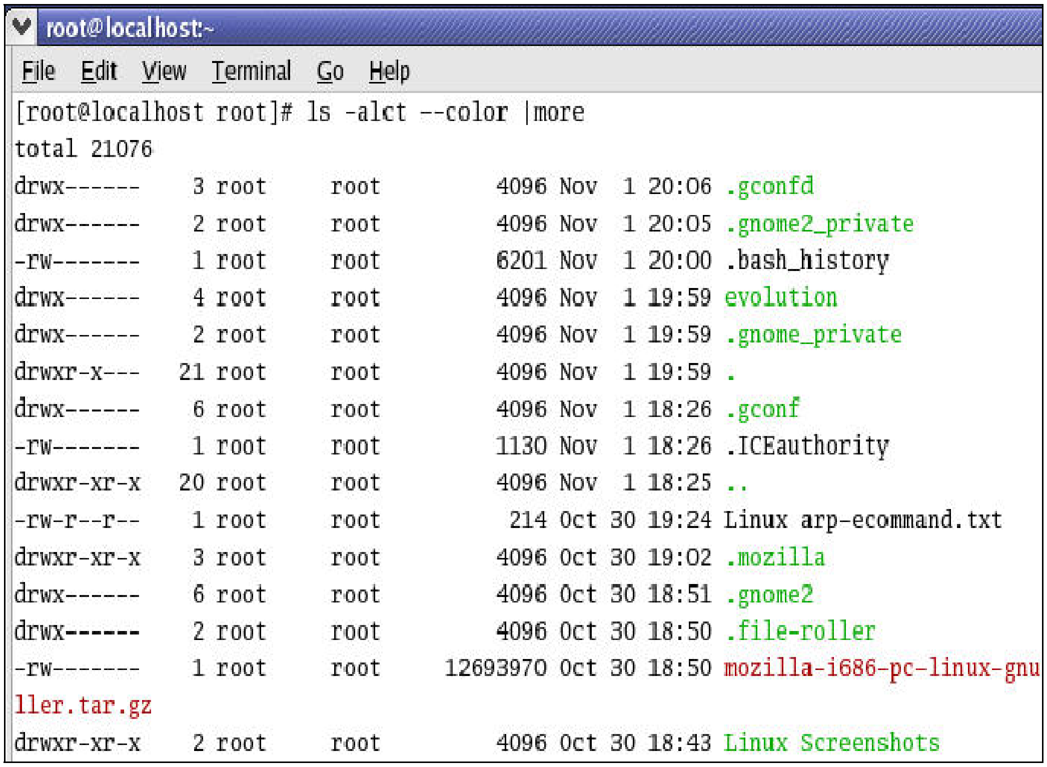
```
-rw-r--r--  1  root    root       2823     Nov 1 20:40
```

- The first character in the above line specifies whether the object is a file (-), a directory (d), or a link to another file or directory (l).
- The next group of three characters defines the read, write, and execute permissions (in that order) for the owner of the object.
- The next group of three characters describes the permissions for the group that controls the object.
- The final group describes the permissions for everyone else, usually called the global permissions.
- The owner of the object is given in the third column and the group the object belongs to is in the fourth column.

In Figure 36, root owns anaconda-ks.cfg and has read and write access. The group and everyone else can only read the file. Remember that root always has the ability to change the permissions on any file or directory.

To display all files sorted by the last modification time and date of each file, enter the following command:

```
ls – alct
```



```
[root@localhost root]# ls -alct --color |more
total 21076
drwx------    3 root     root         4096 Nov  1 20:06 .gconfd
drwx------    2 root     root         4096 Nov  1 20:05 .gnome2_private
-rw-------    1 root     root         6201 Nov  1 20:00 .bash_history
drwx------    4 root     root         4096 Nov  1 19:59 evolution
drwx------    2 root     root         4096 Nov  1 19:59 .gnome_private
drwxr-x---   21 root     root         4096 Nov  1 19:59 .
drwx------    6 root     root         4096 Nov  1 18:26 .gconf
-rw-------    1 root     root         1130 Nov  1 18:26 .ICEauthority
drwxr-xr-x   20 root     root         4096 Nov  1 18:25 ..
-rw-r--r--    1 root     root          214 Oct 30 19:24 Linux arp-ecommand.txt
drwxr-xr-x    3 root     root         4096 Oct 30 19:02 .mozilla
drwx------    6 root     root         4096 Oct 30 18:51 .gnome2
drwx------    2 root     root         4096 Oct 30 18:50 .file-roller
-rw-------    1 root     root     12693970 Oct 30 18:50 mozilla-i686-pc-linux-gnu
ller.tar.gz
drwxr-xr-x    2 root     root         4096 Oct 30 18:43 Linux Screenshots
```

Figure 37: Using ls – alct *to Show Last Modification Date and Time*

Common *ls* parameters are listed in Table 7.

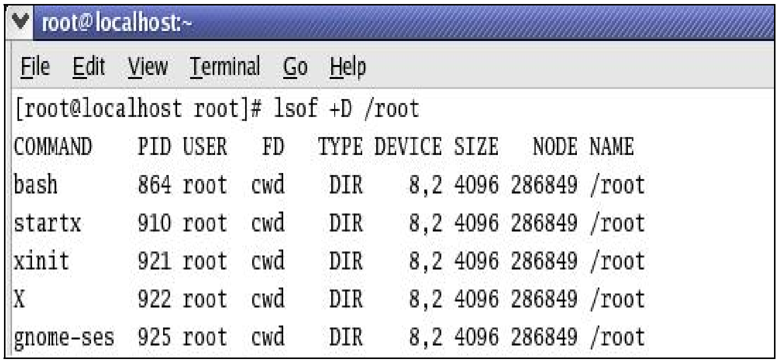
Table 7: Common ls *Parameters*

Parameter	**Function**
-a	Do not hide entries starting with a.
-l	Use a long listing format.
-R	List subdirectories recursively
-u	Show last access time
-c	Show last modification time
-t	Sort list by time

Lsof (Linux)

The native *lsof* command allows you to list information about open files and the processes that might have launched them. You can discover, among other things, the command and user that opened the file as well as the file's size and PID.

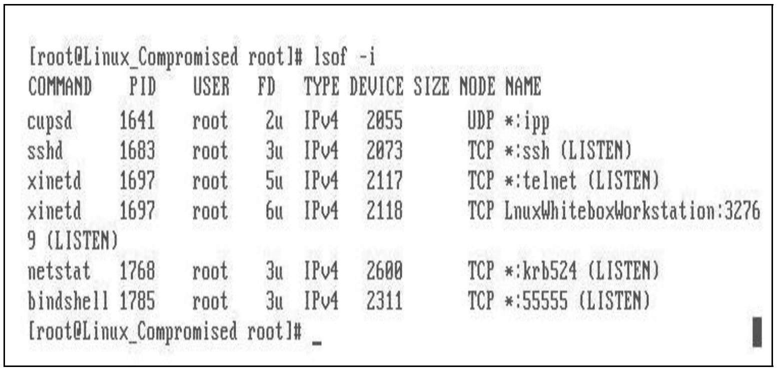
The *+D [directory]* parameter shown in Figure 38 forces *lsof* to perform a full-descent directory tree search for open files within a specific directory.



```
[root@localhost root]# lsof +D /root
COMMAND    PID USER   FD   TYPE DEVICE SIZE   NODE NAME
bash       864 root  cwd    DIR    8,2 4096 286849 /root
startx     910 root  cwd    DIR    8,2 4096 286849 /root
xinit      921 root  cwd    DIR    8,2 4096 286849 /root
X          922 root  cwd    DIR    8,2 4096 286849 /root
gnome-ses  925 root  cwd    DIR    8,2 4096 286849 /root
```

Figure 38: lsof +D

The –*i* option displays all Internet and x.25 (HP-UX) network files.

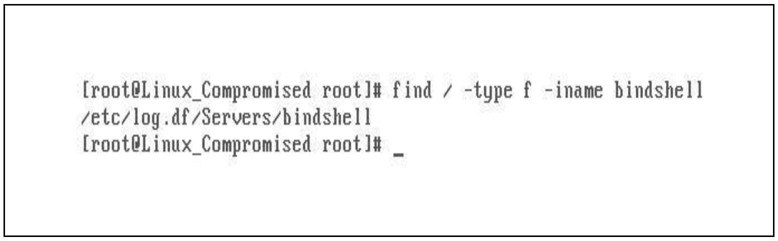


```
[root@Linux_Compromised root]# lsof -i
COMMAND    PID    USER   FD   TYPE DEVICE SIZE NODE NAME
cupsd     1641    root   2u   IPv4   2055       UDP *:ipp
sshd      1683    root   3u   IPv4   2073       TCP *:ssh (LISTEN)
xinetd    1697    root   5u   IPv4   2117       TCP *:telnet (LISTEN)
xinetd    1697    root   6u   IPv4   2118       TCP LnuxWhiteboxWorkstation:3276
9 (LISTEN)
netstat   1768    root   3u   IPv4   2600       TCP *:krb524 (LISTEN)
bindshell 1785    root   3u   IPv4   2311       TCP *:55555 (LISTEN)
[root@Linux_Compromised root]# _
```

Figure 39: lsof -i

Find **and** ***Locate*** **(Linux)**

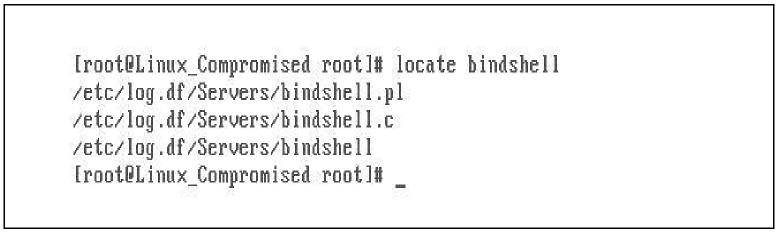
The *find* [directory] [option] [filename] command searches for particular filenames as well.



```
[root@Linux_Compromised root]# find / -type f -iname bindshell
/etc/log.df/Servers/bindshell
[root@Linux_Compromised root]# _
```

Figure 40: The find *Command*

The *locate* [filename] command searches the complete file system for references to the filename you specify.



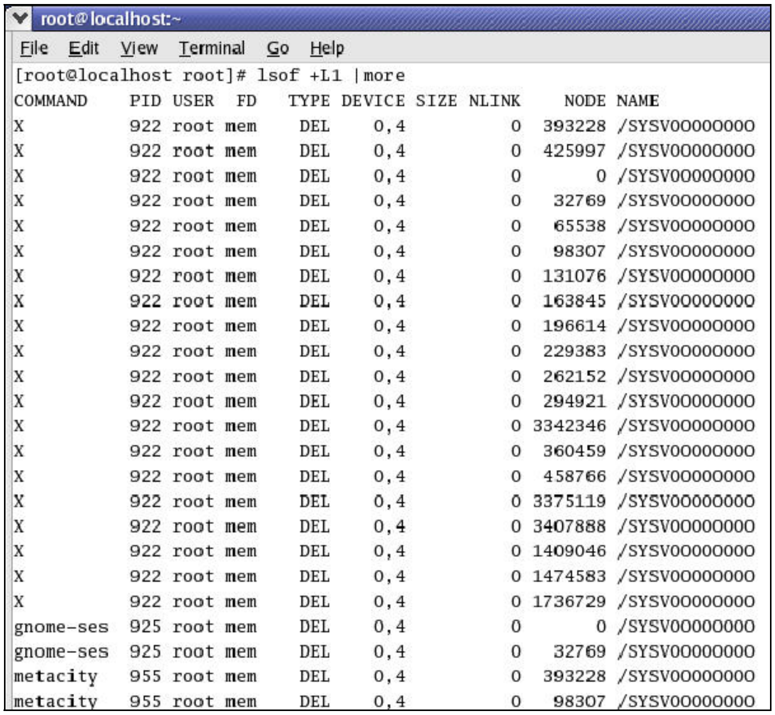
```
[root@Linux_Compromised root]# locate bindshell
/etc/log.df/Servers/bindshell.pl
/etc/log.df/Servers/bindshell.c
/etc/log.df/Servers/bindshell
[root@Linux_Compromised root]# _
```

Figure 41: The locate *Command*

Unlinked Files (Linux)

Unlinked files are marked for deletion when processes that access them terminate [Mandia 01]. Because they disappear when the system is powered down, you should attempt to recover them.

To display open files that have been unlinked and marked for deletion, use the lsof +L1 command as shown in Figure 42.



```
[root@localhost root]# lsof +L1 |more
COMMAND     PID USER   FD   TYPE DEVICE SIZE NLINK    NODE NAME
X           922 root mem    DEL    0,4           0  393228 /SYSV00000000
X           922 root mem    DEL    0,4           0  425997 /SYSV00000000
X           922 root mem    DEL    0,4           0       0 /SYSV00000000
X           922 root mem    DEL    0,4           0   32769 /SYSV00000000
X           922 root mem    DEL    0,4           0   65538 /SYSV00000000
X           922 root mem    DEL    0,4           0   98307 /SYSV00000000
X           922 root mem    DEL    0,4           0  131076 /SYSV00000000
X           922 root mem    DEL    0,4           0  163845 /SYSV00000000
X           922 root mem    DEL    0,4           0  196614 /SYSV00000000
X           922 root mem    DEL    0,4           0  229383 /SYSV00000000
X           922 root mem    DEL    0,4           0  262152 /SYSV00000000
X           922 root mem    DEL    0,4           0  294921 /SYSV00000000
X           922 root mem    DEL    0,4           0 3342346 /SYSV00000000
X           922 root mem    DEL    0,4           0  360459 /SYSV00000000
X           922 root mem    DEL    0,4           0  458766 /SYSV00000000
X           922 root mem    DEL    0,4           0 3375119 /SYSV00000000
X           922 root mem    DEL    0,4           0 3407888 /SYSV00000000
X           922 root mem    DEL    0,4           0 1409046 /SYSV00000000
X           922 root mem    DEL    0,4           0 1474583 /SYSV00000000
X           922 root mem    DEL    0,4           0 1736729 /SYSV00000000
gnome-ses   925 root mem    DEL    0,4           0       0 /SYSV00000000
gnome-ses   925 root mem    DEL    0,4           0   32769 /SYSV00000000
metacity    955 root mem    DEL    0,4           0  393228 /SYSV00000000
metacity    955 root mem    DEL    0,4           0   98307 /SYSV00000000
```

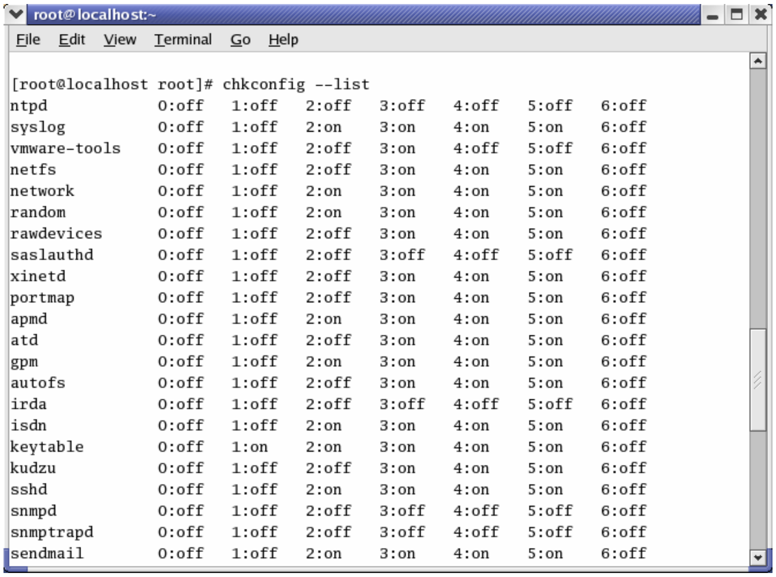
Figure 42: The lsof +L1 *Command*

How UNIX deletes a file is best described by Prosise and Mandia [Mandia 01]:

> *When an attacker runs a process, he usually deletes the program file he executed from the file system in an effort to hide his actions. He is not truly deleting the program on the hard drive. The attacker is unlinking the file.*
>
> *UNIX tracks a file's link count, which is a positive integer representing the number of processes currently using the file. When the link count equals zero, that means no process is using or needs the file, so it will be deleted. When an attacker deletes his rogue program, the program on the hard drive is removed from the directory chain (so it will not be displayed in an ls listing), the link count is decremented by one, and the file's deletion time is set. However, note that the link count does not equal zero until the process terminates.*
>
> *Files marked for deletion (these are the unlinked files) at the time a system is powered down—whether gracefully (through normal shutdown procedures) or not (you pulled the power cord)—will ultimately end up deleted on the system. Let's examine why.*
>
> *When UNIX mounts a file system, a "file system dirty" bit is set. When the operating system goes through a normal shutdown, every process is forced to close. The attacker's process terminates normally, and all file handles are closed. This means that the link count on the deleted file is set to zero. After all processes have exited and other general housekeeping items have been completed, the file system is un-mounted, and the file system dirty bit is cleared.*
>
> *If the OS goes through a traumatic shutdown, the file system is left in an unstable state. Unlinked files may still have false link counts, and the dirty bit remains set. On the next boot-up, the file systems are mounted, and the operating system detects the nonzero value of the dirty bit. Most of the time, the administrator will be forced to wait while the system performs a file system check (fsck). The fsck utility will scan the entire file system for damage. If the utility comes across a file with a positive link count and a deletion time set, it will decrement the link count, rendering the file "deleted." Some versions of fsck will re-link the orphaned file to the lost+found directory, but this is not something that you can rely on.*

Chkconfig (Linux)

The *chkconfig –list* command displays a list of startup services that will be run at the five different runlevels. This information may help you identify a rogue or Malware application that starts at one of the five runlevels.



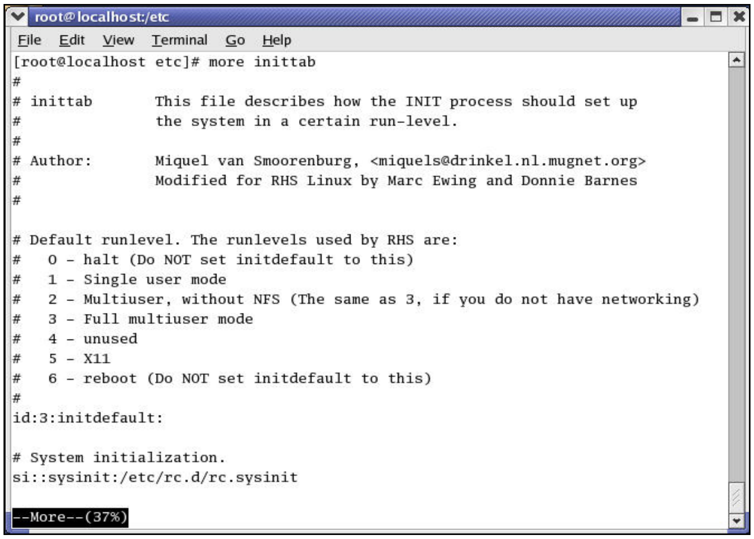
```
[root@localhost root]# chkconfig --list
ntpd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
syslog          0:off   1:off   2:on    3:on    4:on    5:on    6:off
vmware-tools    0:off   1:off   2:off   3:on    4:off   5:off   6:off
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
random          0:off   1:off   2:on    3:on    4:on    5:on    6:off
rawdevices      0:off   1:off   2:off   3:on    4:on    5:on    6:off
saslauthd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
xinetd          0:off   1:off   2:off   3:on    4:on    5:on    6:off
portmap         0:off   1:off   2:off   3:on    4:on    5:on    6:off
apmd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
atd             0:off   1:off   2:off   3:on    4:on    5:on    6:off
gpm             0:off   1:off   2:on    3:on    4:on    5:on    6:off
autofs          0:off   1:off   2:off   3:on    4:on    5:on    6:off
irda            0:off   1:off   2:off   3:off   4:off   5:off   6:off
isdn            0:off   1:off   2:on    3:on    4:on    5:on    6:off
keytable        0:off   1:on    2:on    3:on    4:on    5:on    6:off
kudzu           0:off   1:off   2:off   3:on    4:on    5:on    6:off
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
snmpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
snmptrapd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
sendmail        0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Figure 43: The chkconfig –list *Command*

> ***chkconfig** has five distinct functions: adding new services for management, removing services from management, listing the current startup information for services, changing the startup information for services, and checking the startup state of a particular service.*
>
> *When **chkconfig** is run without any options, it displays usage information. If only a service name is given, it checks to see if the service is configured to be started in the current runlevel. If it is, chkconfig returns true; otherwise it returns false. The **--**level option may be used to have **chkconfig** query an alternative runlevel rather than the current one [Haas 04].*

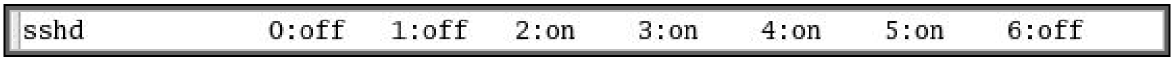
Inittab File (Linux)

The inittab file displays and describes the five runlevels. It describes which processes are started at bootup and during normal operation. Valid runlevels are 0-6 and as we can see in Figure 44, it is set to initdefault setting of 3.



```
[root@localhost etc]# more inittab
#
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modified for RHS Linux by Marc Ewing and Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

--More--(37%)
```

Figure 44: The Inittab File

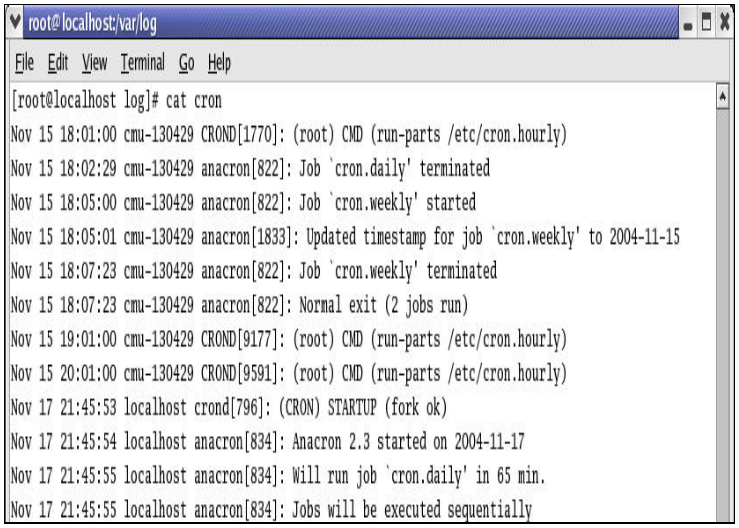


```
sshd              0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

In the above example, the *sshd* service is off at the halt, single user, and reboot runlevels and on at multiuser without NFS, full mulituser, unused, and X11 runlevels.

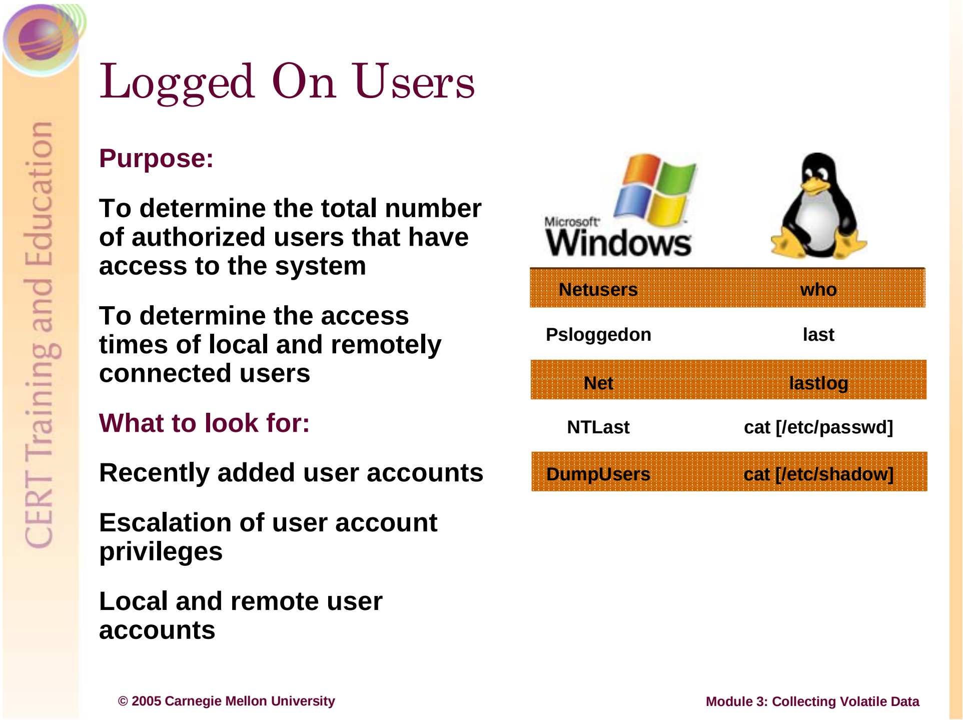
Cron Logs (Linux)

In addition to startup services and open applications and files, collect the currently scheduled tasks. Attackers often schedule a malicious file to execute periodically so that the malware remains existent. The cron feature allows system administrators to schedule programs for future execution. All executed cron jobs are logged, usually in the /var/cron/log or in the default logging directory in a file called cron.



```
[root@localhost log]# cat cron
Nov 15 18:01:00 cmu-130429 CROND[1770]: (root) CMD (run-parts /etc/cron.hourly)
Nov 15 18:02:29 cmu-130429 anacron[822]: Job `cron.daily' terminated
Nov 15 18:05:00 cmu-130429 anacron[822]: Job `cron.weekly' started
Nov 15 18:05:01 cmu-130429 anacron[1833]: Updated timestamp for job `cron.weekly' to 2004-11-15
Nov 15 18:07:23 cmu-130429 anacron[822]: Job `cron.weekly' terminated
Nov 15 18:07:23 cmu-130429 anacron[822]: Normal exit (2 jobs run)
Nov 15 19:01:00 cmu-130429 CROND[9177]: (root) CMD (run-parts /etc/cron.hourly)
Nov 15 20:01:00 cmu-130429 CROND[9591]: (root) CMD (run-parts /etc/cron.hourly)
Nov 17 21:45:53 localhost crond[796]: (CRON) STARTUP (fork ok)
Nov 17 21:45:54 localhost anacron[834]: Anacron 2.3 started on 2004-11-17
Nov 17 21:45:55 localhost anacron[834]: Will run job `cron.daily' in 65 min.
Nov 17 21:45:55 localhost anacron[834]: Jobs will be executed sequentially
```

Figure 45: A Cron Log



CERT Training and Education

Logged On Users

Purpose:

To determine the total number of authorized users that have access to the system

To determine the access times of local and remotely connected users

What to look for:

Recently added user accounts

Escalation of user account privileges

Local and remote user accounts

Microsoft Windows	
Netusers	who
Psloggedon	last
Net	lastlog
NTLast	cat [/etc/passwd]
DumpUsers	cat [/etc/shadow]

Module 3: Collecting Volatile Data

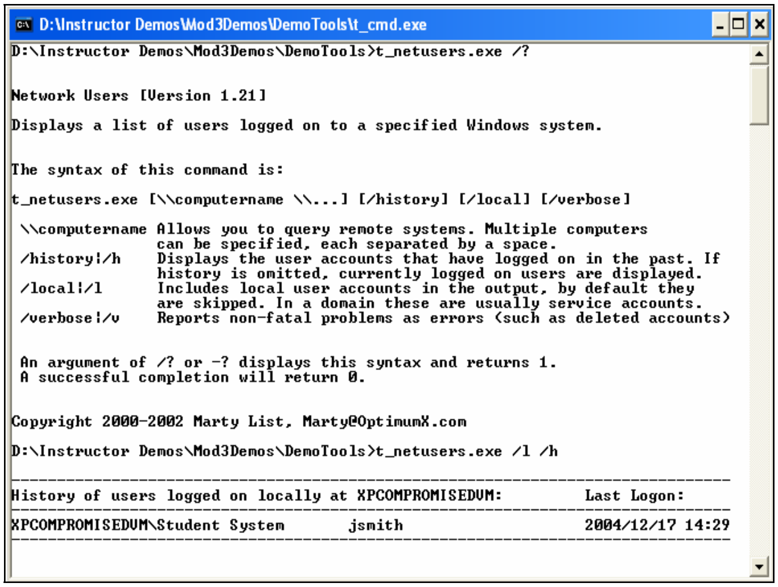
3.9.1.6 Logged On Users

The tools in this section help you identify logged on users. When you approach a suspicious computer, at least one user is generally logged on locally and others may be logged on remotely. Pay attention to the following:

- recently added user accounts
- escalated privileges for existing user accounts
- remote access accounts added to the system (remote users generally access the system through a shared drive, folder, backdoor, or printer)
- the total number of local and remote users that have access to the system or are currently logged onto the system
- times that the users logged on to the system, logged off, and the period during which they remained logged onto the system

Netusers (Windows: Local Users)

SomarSoft's *Netusers*, a freeware command-line utility, allows you to display locally logged on users. It can display current interactive users or all users who have ever logged on through the computer [SystemTools 04].

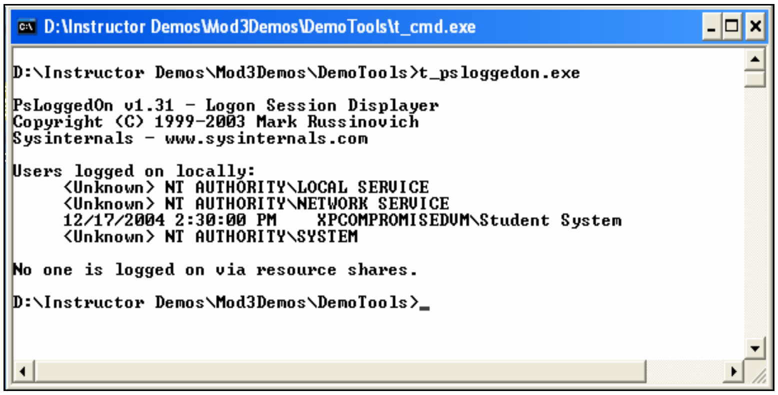


```
D:\Instructor Demos\Mod3Demos\DemoTools>t_netusers.exe /?


Network Users [Version 1.21]

Displays a list of users logged on to a specified Windows system.


The syntax of this command is:

t_netusers.exe [\\computername \\...] [/history] [/local] [/verbose]

 \\computername Allows you to query remote systems. Multiple computers
                can be specified, each separated by a space.
 /history|/h    Displays the user accounts that have logged on in the past. If
                history is omitted, currently logged on users are displayed.
 /local|/l      Includes local user accounts in the output, by default they
                are skipped. In a domain these are usually service accounts.
 /verbose|/v    Reports non-fatal problems as errors (such as deleted accounts)


 An argument of /? or -? displays this syntax and returns 1.
 A successful completion will return 0.


Copyright 2000-2002 Marty List, Marty@OptimumX.com

D:\Instructor Demos\Mod3Demos\DemoTools>t_netusers.exe /l /h

-------------------------------------------------------------------------------
History of users logged on locally at XPCOMPROMISEDVM:          Last Logon:
-------------------------------------------------------------------------------
XPCOMPROMISEDVM\Student System       jsmith                     2004/12/17 14:29
-------------------------------------------------------------------------------
```

Figure 46: Netusers

PsLoggedOn (Windows: Local and Remote Users)

Sysinternals' *PsLoggedOn*, a freeware utility, allows you to display locally and remotely logged on users. In Figure 47, the machine was logged on to at 2:30 p.m. and no user is logged on via resource shares.

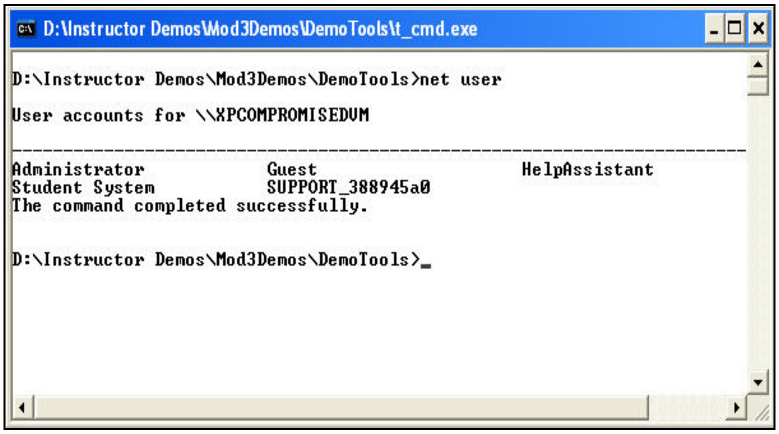


```
D:\Instructor Demos\Mod3Demos\DemoTools>t_psloggedon.exe

PsLoggedOn v1.31 - Logon Session Displayer
Copyright (C) 1999-2003 Mark Russinovich
Sysinternals - www.sysinternals.com

Users logged on locally:
     <Unknown> NT AUTHORITY\LOCAL SERVICE
     <Unknown> NT AUTHORITY\NETWORK SERVICE
     12/17/2004 2:30:00 PM     XPCOMPROMISEDVM\Student System
     <Unknown> NT AUTHORITY\SYSTEM

No one is logged on via resource shares.

D:\Instructor Demos\Mod3Demos\DemoTools>_
```

Figure 47: PsLoggedOn

Net (Windows: Local and Remote Users)

The native *net [user]* command displays all local and remote users. Generally, system administrators use *net user* to create and modify user accounts on computers. If they use the command without command-line switches, all user accounts for the computer are displayed.



```
D:\Instructor Demos\Mod3Demos\DemoTools>net user

User accounts for \\XPCOMPROMISEDVM

-------------------------------------------------------------------------------
Administrator            Guest                    HelpAssistant
Student System           SUPPORT_388945a0
The command completed successfully.


D:\Instructor Demos\Mod3Demos\DemoTools>_
```

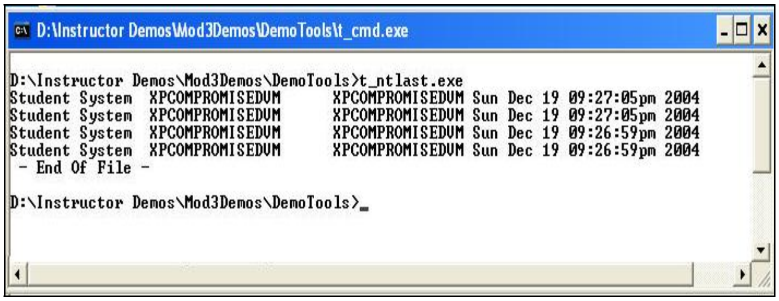
Figure 48: The net user *Command*

Net (Windows: Remote Users)

The native *net [session]* command displays information about remote connections established with the suspicious computer. For each connection, it displays the remote computer's name, IP address, client type, and idle time.

NTLast (Windows: Remote Users)

The *NTLast* utility from Foundstone can be used to collect event logs associated with logon attempts on the suspicious computer.



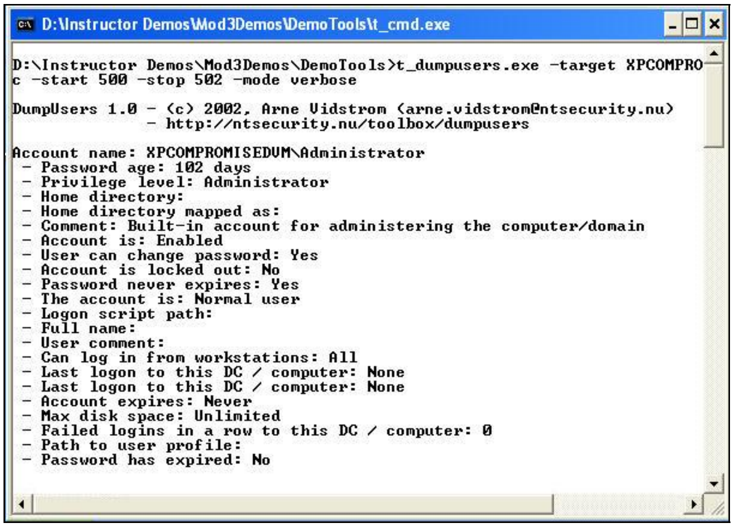
```
D:\Instructor Demos\Mod3Demos\DemoTools>t_ntlast.exe
Student System  XPCOMPROMISEDVM      XPCOMPROMISEDVM Sun Dec 19 09:27:05pm 2004
Student System  XPCOMPROMISEDVM      XPCOMPROMISEDVM Sun Dec 19 09:27:05pm 2004
Student System  XPCOMPROMISEDVM      XPCOMPROMISEDVM Sun Dec 19 09:26:59pm 2004
Student System  XPCOMPROMISEDVM      XPCOMPROMISEDVM Sun Dec 19 09:26:59pm 2004
 - End Of File -

D:\Instructor Demos\Mod3Demos\DemoTools>_
```

Figure 49: NTLast

For *NTLast* to work, the audit log feature had to have been enabled through the suspicious computer's Local Security Policy. If it is not, you will not get any information.

DumpUsers (Windows: Local Users)

NTSecurity.nu’s *DumpUsers* utility dumps account names and information even though RestrictAnonymous has been set to 1.



```
D:\Instructor Demos\Mod3Demos\DemoTools>t_dumpusers.exe -target XPCOMPRO
c -start 500 -stop 502 -mode verbose

DumpUsers 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
              - http://ntsecurity.nu/toolbox/dumpusers

Account name: XPCOMPROMISEDVM\Administrator
 - Password age: 102 days
 - Privilege level: Administrator
 - Home directory:
 - Home directory mapped as:
 - Comment: Built-in account for administering the computer/domain
 - Account is: Enabled
 - User can change password: Yes
 - Account is locked out: No
 - Password never expires: Yes
 - The account is: Normal user
 - Logon script path:
 - Full name:
 - User comment:
 - Can log in from workstations: All
 - Last logon to this DC / computer: None
 - Last logon to this DC / computer: None
 - Account expires: Never
 - Max disk space: Unlimited
 - Failed logins in a row to this DC / computer: 0
 - Path to user profile:
 - Password has expired: No
```

Figure 50: DumpUsers

The command line options selected in Figure 50 are as follows:

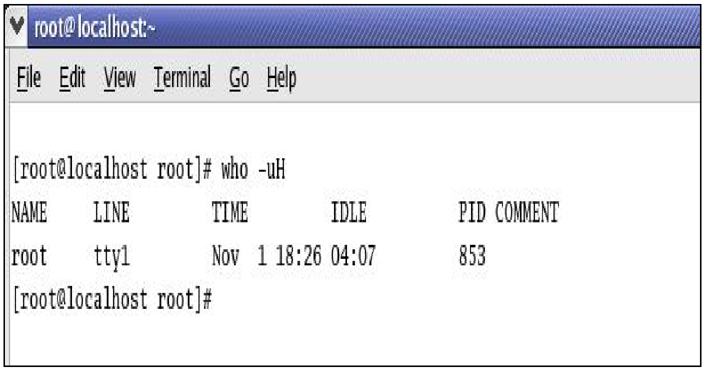
- Target (suspicious computer): XPCOMPROMISEDVM
- Type: notdc (not a domain controller)
- Start: 500 (the RID from which to start collecting information)
- Stop: 502 (the RID from which to stop)
- Mode: verbose

Who (Linux: Local Users)

The native *who* command displays the user that is currently logged on locally. With no options, *who* lists the name of each user currently logged in with their terminal, the time they logged on, and the name of the host from which they have logged in. Supply an optional system file (default is */etc/utmp*) to obtain additional information.

Who Am I, Who –uH (Linux: Local Users)

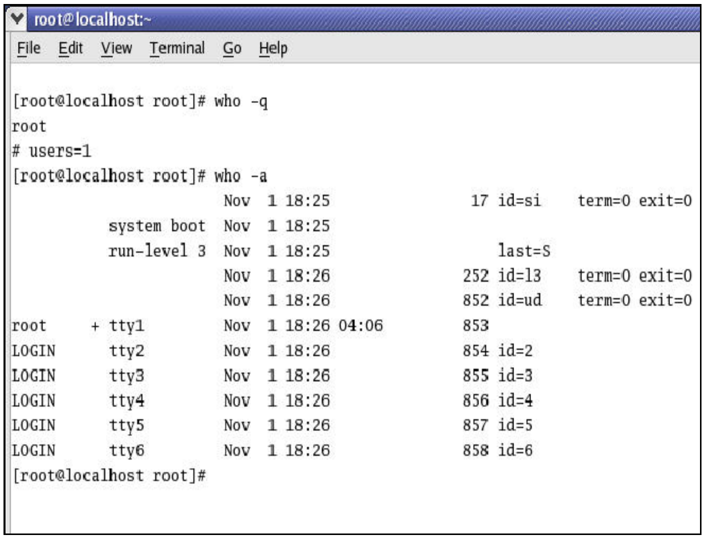
The native *who am i* command lets you quickly determine the currently logged on user. The *who –uH* command displays the idle times for logged on user(s). Idle times can be good indicators of a logged on user’s previous activity.



```
[root@localhost root]# who -uH
NAME     LINE         TIME         IDLE          PID COMMENT
root     tty1         Nov  1 18:26 04:07         853
[root@localhost root]#
```

Figure 51: *The* who –uH *Command*

The native *who –q* command displays all logged in user names and the number of logged in users. Without any parameters, the command displays the time of the last system boot and users currently logged on. It also prints any dead processes, system login processes, active processes spawned by init, the current run level, and the last system clock change.



```
[root@localhost root]# who -q
root
# users=1
[root@localhost root]# who -a
                        Nov  1 18:25              17 id=si    term=0 exit=0
           system boot  Nov  1 18:25
           run-level 3  Nov  1 18:25                 last=S
                        Nov  1 18:26             252 id=l3    term=0 exit=0
                        Nov  1 18:26             852 id=ud    term=0 exit=0
root     + tty1         Nov  1 18:26 04:06       853
LOGIN      tty2         Nov  1 18:26             854 id=2
LOGIN      tty3         Nov  1 18:26             855 id=3
LOGIN      tty4         Nov  1 18:26             856 id=4
LOGIN      tty5         Nov  1 18:26             857 id=5
LOGIN      tty6         Nov  1 18:26             858 id=6
[root@localhost root]#
```

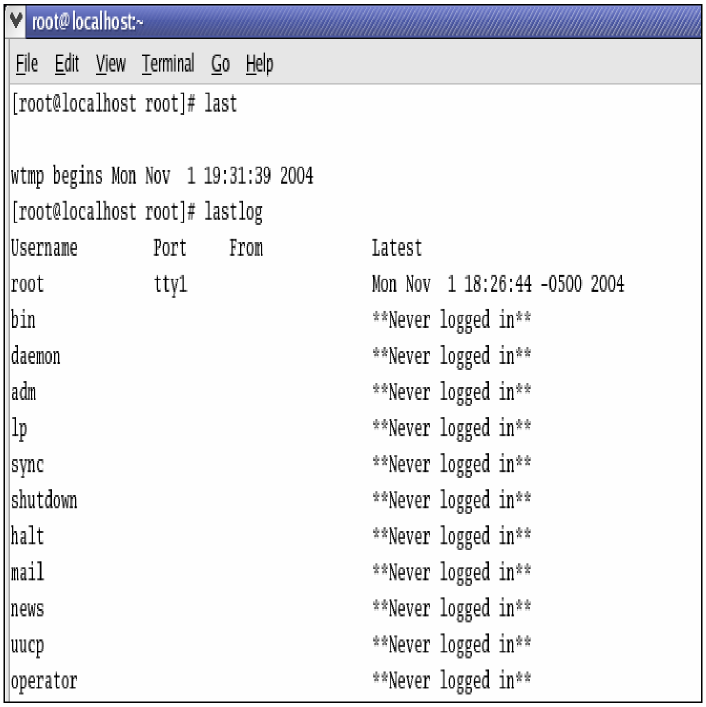
Figure 52: *The* who –q *Command*

Who –all/–a **(Linux: Local and Remote Users)**

The native *who –all* or *who –a* commands display all currently logged on users, local and remote.

Last (Linux: Local and Remote Users)

The native *last* command displays a history of logged on users, local and remote. By default, recent logins from the */var/log/wtmp* file are included. Specify a tty number or username for user- or terminal-specific output.



```
[root@localhost root]# last


wtmp begins Mon Nov  1 19:31:39 2004
[root@localhost root]# lastlog
Username         Port     From             Latest
root             tty1                      Mon Nov  1 18:26:44 -0500 2004
bin                                        **Never logged in**
daemon                                     **Never logged in**
adm                                        **Never logged in**
lp                                         **Never logged in**
sync                                       **Never logged in**
shutdown                                   **Never logged in**
halt                                       **Never logged in**
mail                                       **Never logged in**
news                                       **Never logged in**
uucp                                       **Never logged in**
operator                                   **Never logged in**
```

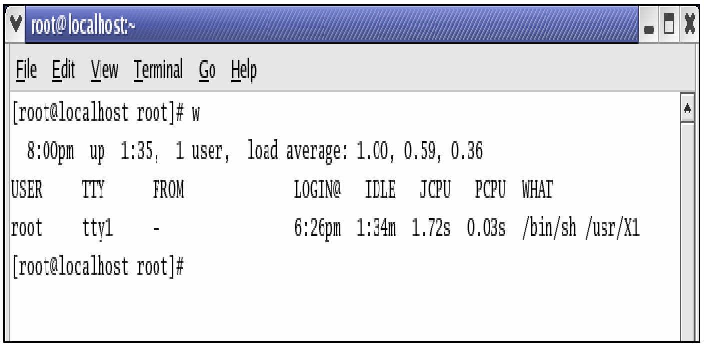
Figure 53: The last *Command*

Lastlog (Linux: Local and Remote Users)

The native *lastlog* command displays the last login times for system accounts. Login information is read from the */var/log/lastlog* file.

W (Linux: Local and Remote Users)

The native *w* command displays summaries of system usage, currently logged on users, and logged on user activities. This command is a combination of *uptime*, *who*, and *ps -a*. Specify a user for user-specific details.

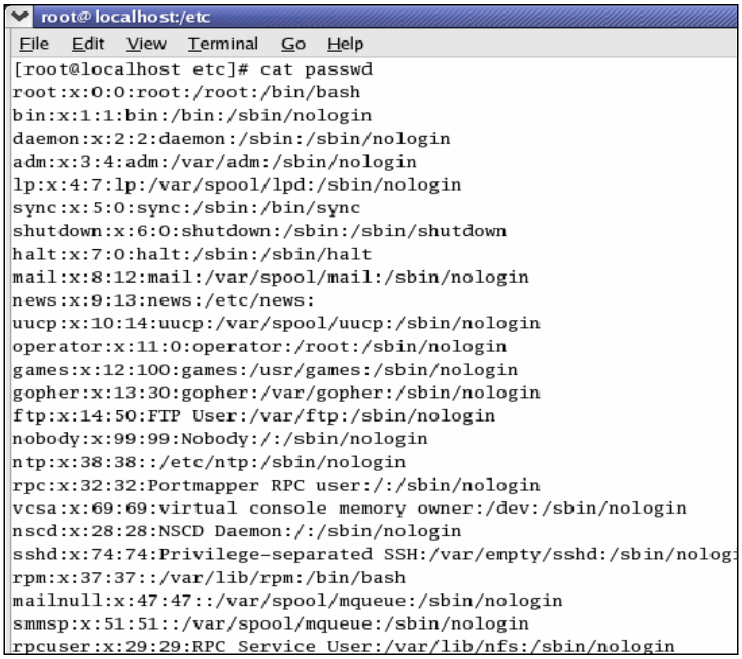


```
[root@localhost root]# w
  8:00pm  up  1:35,  1 user,  load average: 1.00, 0.59, 0.36
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
root     tty1     -                 6:26pm  1:34m  1.72s  0.03s  /bin/sh /usr/X1
[root@localhost root]#
```

Figure 54: The w *Command*

Passwd (Linux: Local and Remote Users)

The */etc/passwd* text file contains user account information, including one-way encrypted passwords. To display the file, use the *cat /etc/passwd* command.

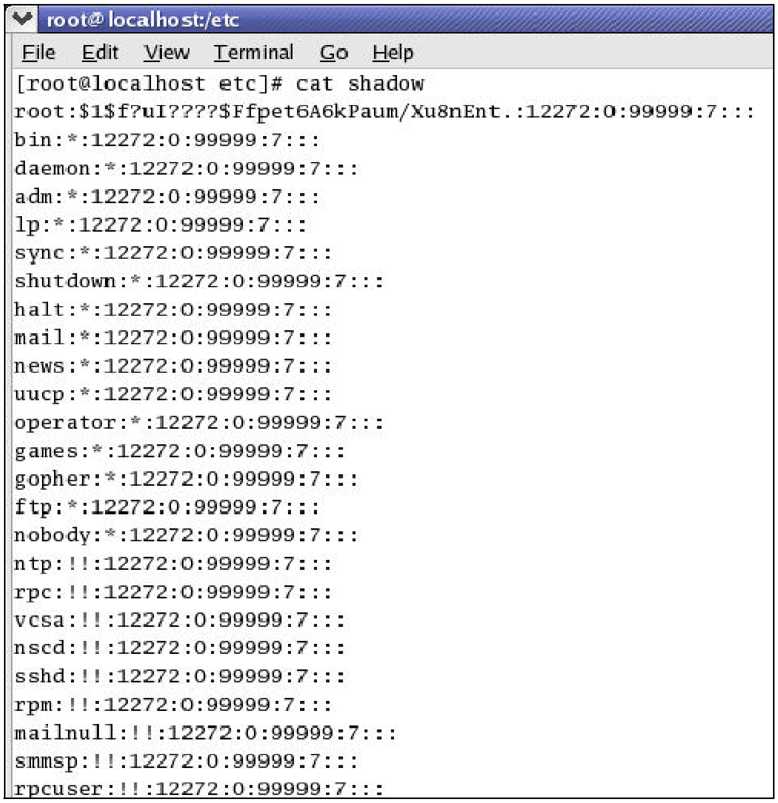


```
[root@localhost etc]# cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nolog
rpm:x:37:37::/var/lib/rpm:/bin/bash
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
```

Figure 55: The cat /etc/passwd *Command*

Shadow (Linux: Local and Remote Users)

The */etc/shadow* text file contains password and optional password aging information and is readable only by the root user.

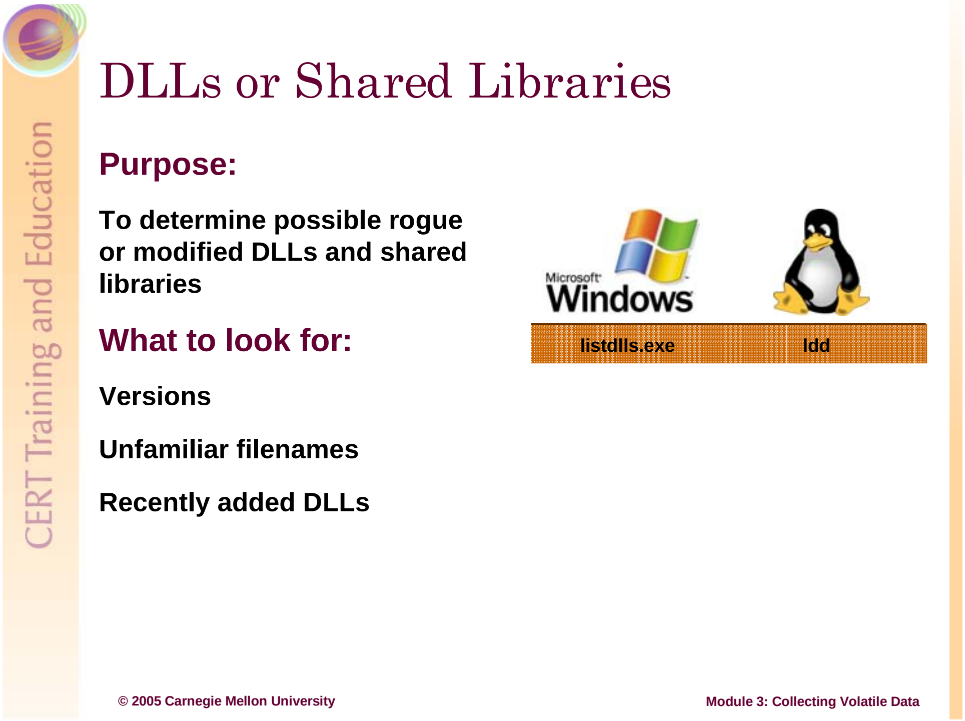


```
[root@localhost etc]# cat shadow
root:$1$f?uI????$Ffpet6A6kPaum/Xu8nEnt.:12272:0:99999:7:::
bin:*:12272:0:99999:7:::
daemon:*:12272:0:99999:7:::
adm:*:12272:0:99999:7:::
lp:*:12272:0:99999:7:::
sync:*:12272:0:99999:7:::
shutdown:*:12272:0:99999:7:::
halt:*:12272:0:99999:7:::
mail:*:12272:0:99999:7:::
news:*:12272:0:99999:7:::
uucp:*:12272:0:99999:7:::
operator:*:12272:0:99999:7:::
games:*:12272:0:99999:7:::
gopher:*:12272:0:99999:7:::
ftp:*:12272:0:99999:7:::
nobody:*:12272:0:99999:7:::
ntp:!!:12272:0:99999:7:::
rpc:!!:12272:0:99999:7:::
vcsa:!!:12272:0:99999:7:::
nscd:!!:12272:0:99999:7:::
sshd:!!:12272:0:99999:7:::
rpm:!!:12272:0:99999:7:::
mailnull:!!:12272:0:99999:7:::
smmsp:!!:12272:0:99999:7:::
rpcuser:!!:12272:0:99999:7:::
```

Figure 56: The cat shadow *Command*

As in the */etc/passwd* file, each user's information is on a separate line. Each line has nine fields, delineated by colons:

- username
- encrypted password
- date password last changed
- number of days before password can be changed
- number of days before password can be changed
- number of days before password change is required
- number of days warning before password change
- number of days before the account is disabled
- date since the account has been disabled



CERT Training and Education

DLLs or Shared Libraries

Purpose:

To determine possible rogue or modified DLLs and shared libraries

What to look for:

Versions

Unfamiliar filenames

Recently added DLLs

listdlls.exe	ldd

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.9.1.7 DLLs or Shared Libraries

A DLL (or shared library) file contains functions that are compiled, linked, and saved separately from the processes that use them. Functions in DLLs can be used by more than one running process. The OS maps the DLLs into the process’ address space when the process is started or while it is running.

The tools in this section help you identify currently loaded DLLs or shared libraries so that you can examine them and determine if any are rogue. Filenames for rogue DLLs may be unidentified or unfamiliar (best case) or they may be masked so that they appear to be legitimate even though they have been corrupted or trigger malicious activity when linked.

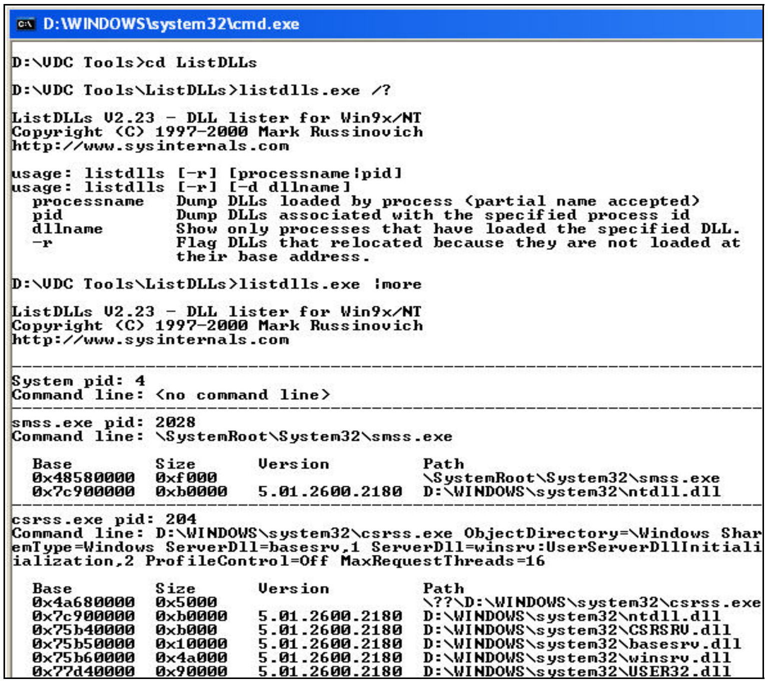
As you use the tools in this section to identify DLLs, pay attention to the following:

- processes that are using the running DLLs
- unfamiliar DLLs
- compare the list of DLLs for a process with the expected DLLs for that process (check for number of DLLs and their dates, which should be the OS install date unless patches and hot fixes have been applied)

ListDLLs (Windows)

Sysinternals' *ListDLLs* freeware utility displays all loaded DLLs with their version numbers, associates each loaded DLL with its application or executable, and flags loaded DLLs that have different version numbers than their corresponding on-disk files.

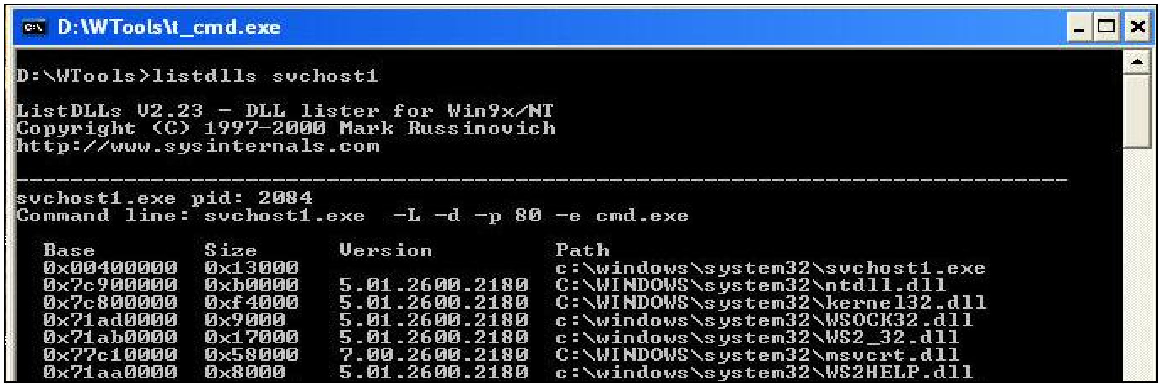
In Figure 57, *ListDLLs* is run without options to produce a list of DLLs.



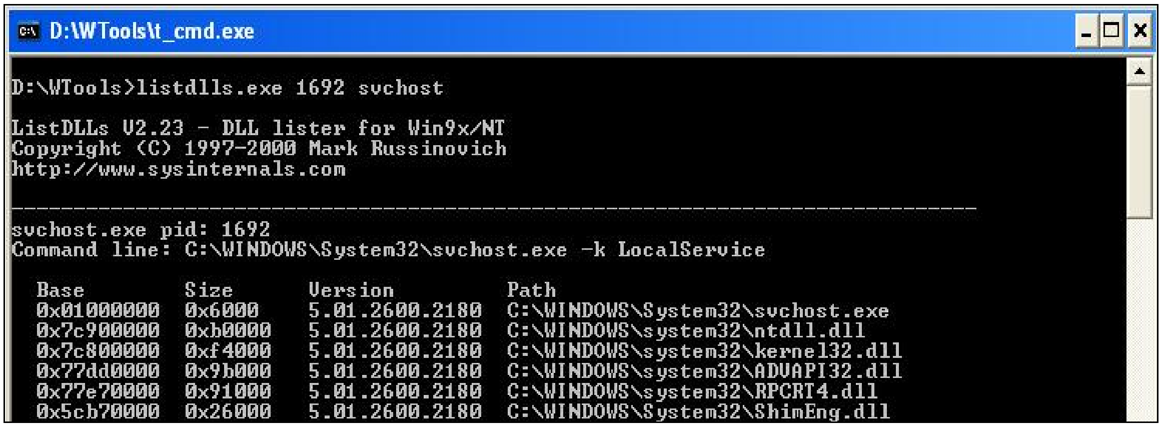
```
D:\VDC Tools>cd ListDLLs

D:\VDC Tools\ListDLLs>listdlls.exe /?

ListDLLs V2.23 - DLL lister for Win9x/NT
Copyright (C) 1997-2000 Mark Russinovich
http://www.sysinternals.com

usage: listdlls [-r] [processname|pid]
usage: listdlls [-r] [-d dllname]
  processname   Dump DLLs loaded by process (partial name accepted)
  pid           Dump DLLs associated with the specified process id
  dllname       Show only processes that have loaded the specified DLL.
  -r            Flag DLLs that relocated because they are not loaded at
                their base address.

D:\VDC Tools\ListDLLs>listdlls.exe |more

ListDLLs V2.23 - DLL lister for Win9x/NT
Copyright (C) 1997-2000 Mark Russinovich
http://www.sysinternals.com

------------------------------------------------------------------------------
System pid: 4
Command line: <no command line>
------------------------------------------------------------------------------
smss.exe pid: 2028
Command line: \SystemRoot\System32\smss.exe

  Base        Size      Version          Path
  0x48580000  0xf000                     \SystemRoot\System32\smss.exe
  0x7c900000  0xb0000   5.01.2600.2180   D:\WINDOWS\system32\ntdll.dll
------------------------------------------------------------------------------
csrss.exe pid: 204
Command line: D:\WINDOWS\system32\csrss.exe ObjectDirectory=\Windows Shar
emType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitiali
ialization,2 ProfileControl=Off MaxRequestThreads=16

  Base        Size      Version          Path
  0x4a680000  0x5000                     \??\D:\WINDOWS\system32\csrss.exe
  0x7c900000  0xb0000   5.01.2600.2180   D:\WINDOWS\system32\ntdll.dll
  0x75b40000  0xb000    5.01.2600.2180   D:\WINDOWS\system32\CSRSRV.dll
  0x75b50000  0x10000   5.01.2600.2180   D:\WINDOWS\system32\basesrv.dll
  0x75b60000  0x4a000   5.01.2600.2180   D:\WINDOWS\system32\winsrv.dll
  0x77d40000  0x90000   5.01.2600.2180   D:\WINDOWS\system32\USER32.dll
```

Figure 57: Using ListDLLs *Without Options to Produce a List of DLLs*

In Figure 58, *ListDLLs* is used to examine a suspected rogue process (*svchost1.exe*). In comparing *svchost1.exe* with the legitimate *svchost.exe*, significant differences in the types and numbers of loaded DLLs are found.



```
D:\WTools>listdlls svchost1

ListDLLs V2.23 - DLL lister for Win9x/NT
Copyright (C) 1997-2000 Mark Russinovich
http://www.sysinternals.com

------------------------------------------------------------------------------
svchost1.exe pid: 2084
Command line: svchost1.exe  -L -d -p 80 -e cmd.exe

  Base        Size      Version          Path
  0x00400000  0x13000                    c:\windows\system32\svchost1.exe
  0x7c900000  0xb0000   5.01.2600.2180   C:\WINDOWS\system32\ntdll.dll
  0x7c800000  0xf4000   5.01.2600.2180   C:\WINDOWS\system32\kernel32.dll
  0x71ad0000  0x9000    5.01.2600.2180   c:\windows\system32\WSOCK32.dll
  0x71ab0000  0x17000   5.01.2600.2180   c:\windows\system32\WS2_32.dll
  0x77c10000  0x58000   7.00.2600.2180   C:\WINDOWS\system32\msvcrt.dll
  0x71aa0000  0x8000    5.01.2600.2180   c:\windows\system32\WS2HELP.dll
```



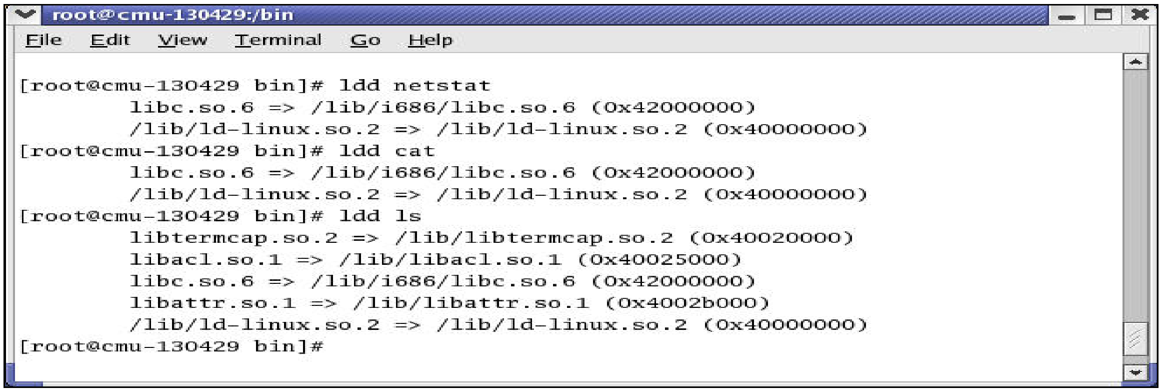
```
D:\WTools>listdlls.exe 1692 svchost

ListDLLs V2.23 - DLL lister for Win9x/NT
Copyright (C) 1997-2000 Mark Russinovich
http://www.sysinternals.com

------------------------------------------------------------------------------
svchost.exe pid: 1692
Command line: C:\WINDOWS\System32\svchost.exe -k LocalService

  Base        Size      Version         Path
  0x01000000  0x6000    5.01.2600.2180  C:\WINDOWS\System32\svchost.exe
  0x7c900000  0xb0000   5.01.2600.2180  C:\WINDOWS\system32\ntdll.dll
  0x7c800000  0xf4000   5.01.2600.2180  C:\WINDOWS\system32\kernel32.dll
  0x77dd0000  0x9b000   5.01.2600.2180  C:\WINDOWS\system32\ADVAPI32.dll
  0x77e70000  0x91000   5.01.2600.2180  C:\WINDOWS\system32\RPCRT4.dll
  0x5cb70000  0x26000   5.01.2600.2180  C:\WINDOWS\System32\ShimEng.dll
```

Figure 58: Using ListDLLs *to Examine a Suspected Rogue Process*

ListDLLs can also be used to display the command lines associated with suspected and legitimate processes. Differences could indicate a rogue process.

Ldd (Linux)

The native *ldd* command displays the shared object files to which an executing binary links. In Linux, each running process or application has loaded or shared libraries.



```
[root@cmu-130429 bin]# ldd netstat
        libc.so.6 => /lib/i686/libc.so.6 (0x42000000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
[root@cmu-130429 bin]# ldd cat
        libc.so.6 => /lib/i686/libc.so.6 (0x42000000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
[root@cmu-130429 bin]# ldd ls
        libtermcap.so.2 => /lib/libtermcap.so.2 (0x40020000)
        libacl.so.1 => /lib/libacl.so.1 (0x40025000)
        libc.so.6 => /lib/i686/libc.so.6 (0x42000000)
        libattr.so.1 => /lib/libattr.so.1 (0x4002b000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
[root@cmu-130429 bin]#
```

Figure 59: The ldd *Command*

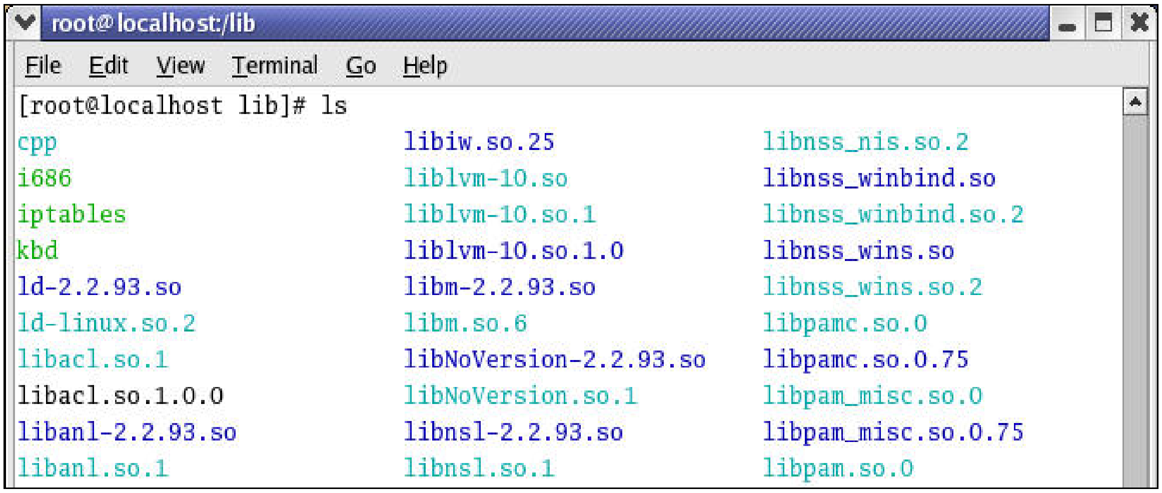
In the following output, *ldd* was used to compare the shared libraries of a known Apache server binary (*httpd*) to a so-called Apache server binary (*apache*). The list of shared library files for the *apache* binary, however, is significantly smaller than the list for *httpd*, which makes *apache* suspect.

```
$ ldd ./apache (Unknown)

        libc.so.6 => /lib/libc.so.6 (0x4001f000)

        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

```
$ ldd /usr/sbin/httpd  (Known)

        libm.so.6 => /lib/libm.so.6 (0x7002c000)

        libcrypt.so.1 => /lib/libcrypt.so.1 (0x700c4000)

        libdb.so.2 => /lib/libdb.so.2 (0x70100000)

        libdb2.so.2 => /lib/libdb2.so.2 (0x70120000)

        libexpat.so.1 => /usr/lib/libexpat.so.1 (0x70180000)

        libdl.so.2 => /lib/libdl.so.2 (0x701b4000)

        libc.so.6 => /lib/libc.so.6 (0x701c8000)

        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x70000000)
```

Ls (Linux)

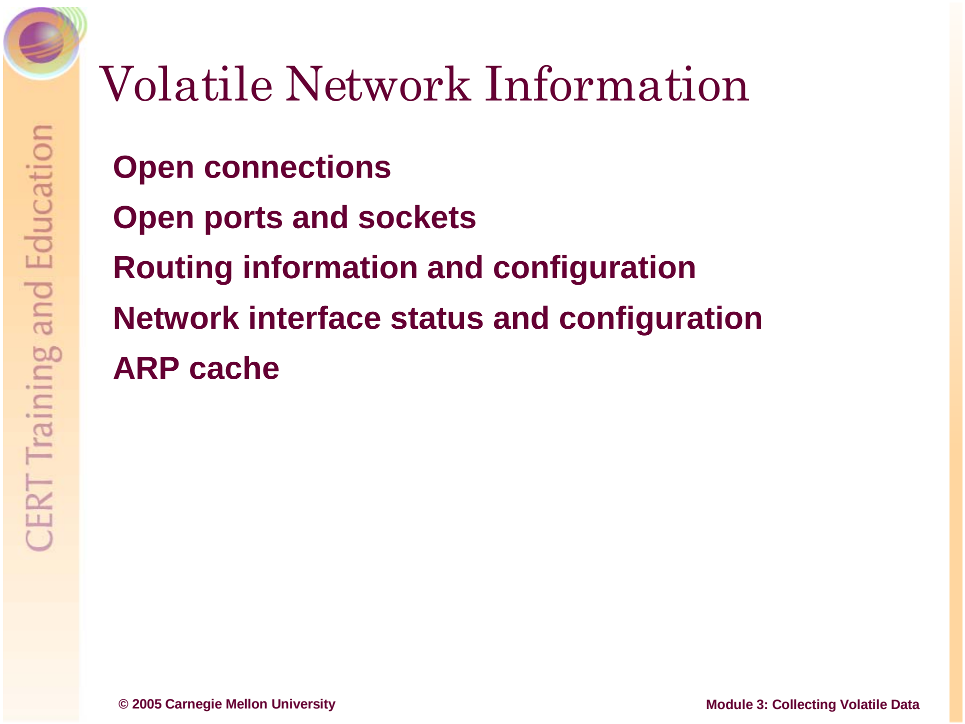
The native *ls* command can be used to display the shared libraries to which each executing binary links. In Figure 60, *ls* is used to display contents of the lib folder.



```
[root@localhost lib]# ls
cpp                 libiw.so.25              libnss_nis.so.2
i686                liblvm-10.so             libnss_winbind.so
iptables            liblvm-10.so.1           libnss_winbind.so.2
kbd                 liblvm-10.so.1.0         libnss_wins.so
ld-2.2.93.so        libm-2.2.93.so           libnss_wins.so.2
ld-linux.so.2       libm.so.6                libpamc.so.0
libacl.so.1         libNoVersion-2.2.93.so   libpamc.so.0.75
libacl.so.1.0.0     libNoVersion.so.1        libpam_misc.so.0
libanl-2.2.93.so    libnsl-2.2.93.so         libpam_misc.so.0.75
libanl.so.1         libnsl.so.1              libpam.so.0
```

Figure 60: The ls *Command*

For more details about default library locations, see Wheeler’s discussion [Wheeler 03].



CERT Training and Education

Volatile Network Information

- Open connections
- Open ports and sockets
- Routing information and configuration
- Network interface status and configuration
- ARP cache

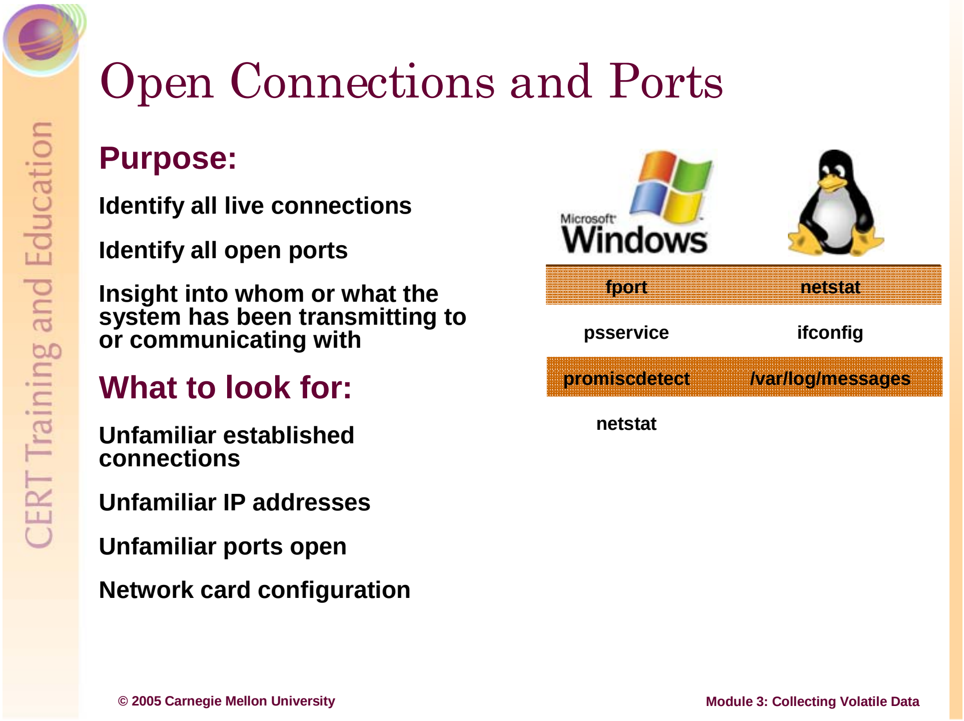
© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.9.2 Volatile Network Information

The types of volatile network information included in this section are not exhaustive, but they are key indictors of the network state that can help you unravel the security incident.

If the machine is standalone or offline, skip this step.



CERT Training and Education

Open Connections and Ports

Purpose:

Identify all live connections

Identify all open ports

Insight into whom or what the system has been transmitting to or communicating with

What to look for:

Unfamiliar established connections

Unfamiliar IP addresses

Unfamiliar ports open

Network card configuration

Microsoft Windows	
fport	netstat
psservice	ifconfig
promiscdetect	/var/log/messages
netstat	

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.9.2.1 Open Connections and Ports

To begin collecting information about open connections and ports, examine the suspicious computer to see if a any type of cable connects it to a network. After you establish all live connections, collect and document the associated IP addresses and open ports. This initial collection of network information may be enough for you to determine whether the security incident was caused remotely or locally.

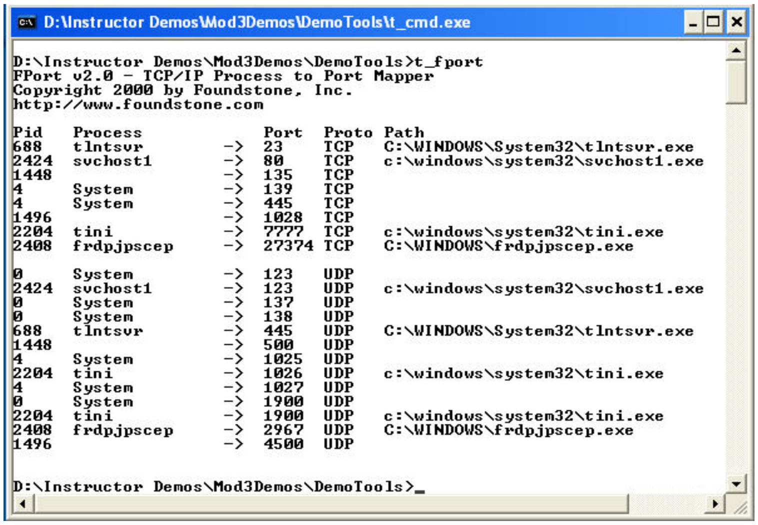
As you use the tools in this section, look for the following:

- unfamiliar IP addresses and ports
- legitimate (established through commands like telnet, ssh, ftp, and netuse) or backdoor connections on abnormal ports
- a promiscuously configured network card interface

Do not expect to know what each open port on the suspicious computer is used for. To look up a port, visit www.treachery.net/tools/ports/lookup.cgi.

Fport (Windows)

The *Fport* freeware utility from Foundstone, like the native *netstat –an* command, displays all open TCP/IP and UDP ports and maps them to the owning application. *Fport* also maps those ports to running processes with the PID, process name, and path.



```
D:\Instructor Demos\Mod3Demos\DemoTools>t_fport
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid   Process            Port  Proto Path
688   tlntsvr        ->  23    TCP   C:\WINDOWS\System32\tlntsvr.exe
2424  svchost1       ->  80    TCP   c:\windows\system32\svchost1.exe
1448                 ->  135   TCP
4     System         ->  139   TCP
4     System         ->  445   TCP
1496                 ->  1028  TCP
2204  tini           ->  7777  TCP   c:\windows\system32\tini.exe
2408  frdpjpscep     ->  27374 TCP   C:\WINDOWS\frdpjpscep.exe

0     System         ->  123   UDP
2424  svchost1       ->  123   UDP   c:\windows\system32\svchost1.exe
0     System         ->  137   UDP
0     System         ->  138   UDP
688   tlntsvr        ->  445   UDP   C:\WINDOWS\System32\tlntsvr.exe
1448                 ->  500   UDP
4     System         ->  1025  UDP
2204  tini           ->  1026  UDP   c:\windows\system32\tini.exe
4     System         ->  1027  UDP
0     System         ->  1900  UDP
2204  tini           ->  1900  UDP   c:\windows\system32\tini.exe
2408  frdpjpscep     ->  2967  UDP   C:\WINDOWS\frdpjpscep.exe
1496                 ->  4500  UDP

D:\Instructor Demos\Mod3Demos\DemoTools>_
```

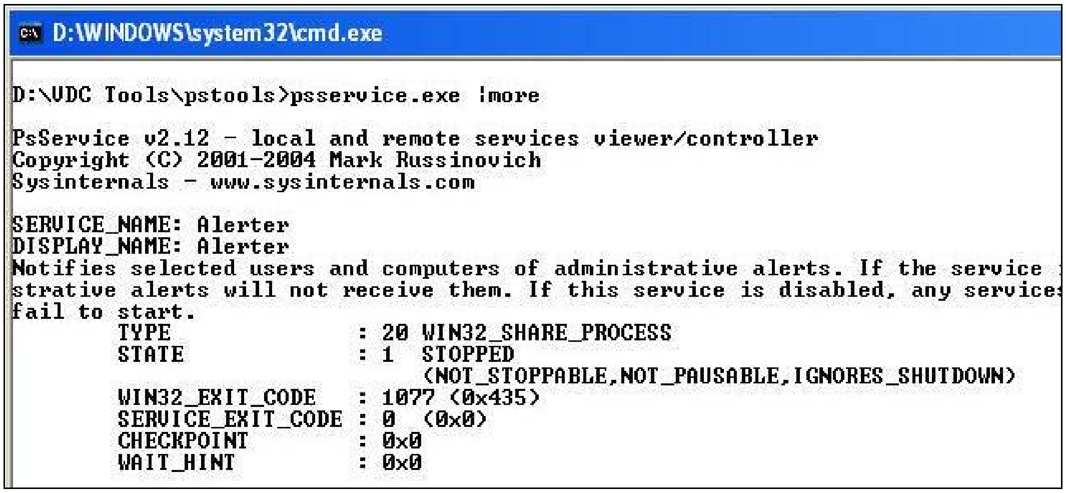
Figure 61: Fport

A GUI-based commercial version of *Fport* is available at http://www.foundstone.com.

PsService (Windows)

Sysinternals' *PsService* freeware utility acts as

> *a service viewer and controller for Windows NT/2K2000. Like the SC utility that's included in the Windows NT and Windows 2000 Resource Kits,* PsService *displays the status, configuration, and dependencies of a service, and allows you to start, stop, pause, resume, and restart them. Unlike the SC utility,* PsService *enables you to log on to a remote system using a different account, for cases when the account from which you run it doesn't have required permissions on the remote system.* PsService *includes a unique service-search capability, which identifies active instances of a service on your network. You would use the search feature if you wanted to locate systems running DHCP servers, for instance [Russinovich 03].*



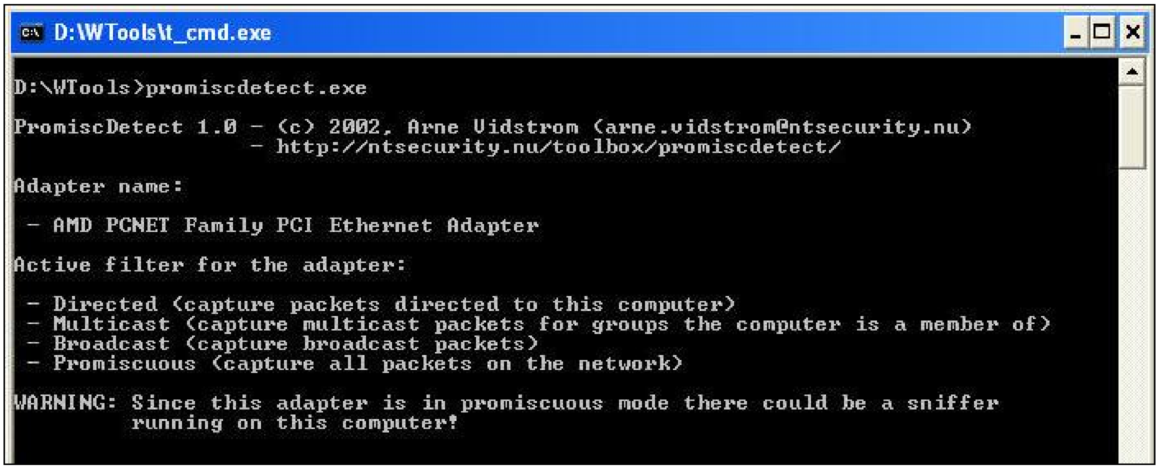
```
D:\VDC Tools\pstools>psservice.exe |more

PsService v2.12 - local and remote services viewer/controller
Copyright (C) 2001-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

SERVICE_NAME: Alerter
DISPLAY_NAME: Alerter
Notifies selected users and computers of administrative alerts. If the service
strative alerts will not receive them. If this service is disabled, any service
fail to start.
        TYPE              : 20 WIN32_SHARE_PROCESS
        STATE             : 1  STOPPED
                               (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE   : 1077 (0x435)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT        : 0x0
        WAIT_HINT         : 0x0
```

Figure 62: PsService

***PromiscDetect* (Windows)**

NTSecurity.nu's freeware utility *PromiscDetect* checks to see if the network adaptor(s) are running in promiscuous mode, which may indicate that a sniffer is running on the suspicious computer. To determine the adapter's mode, look at the *PromiscDetect's* output. If the filters returned are set at Directed, Multicast, and Broadcast, the network card interface is not in promiscuous mode.

PromiscDetect is available at http://ntsecurity.nu/toolbox/promiscdetect.

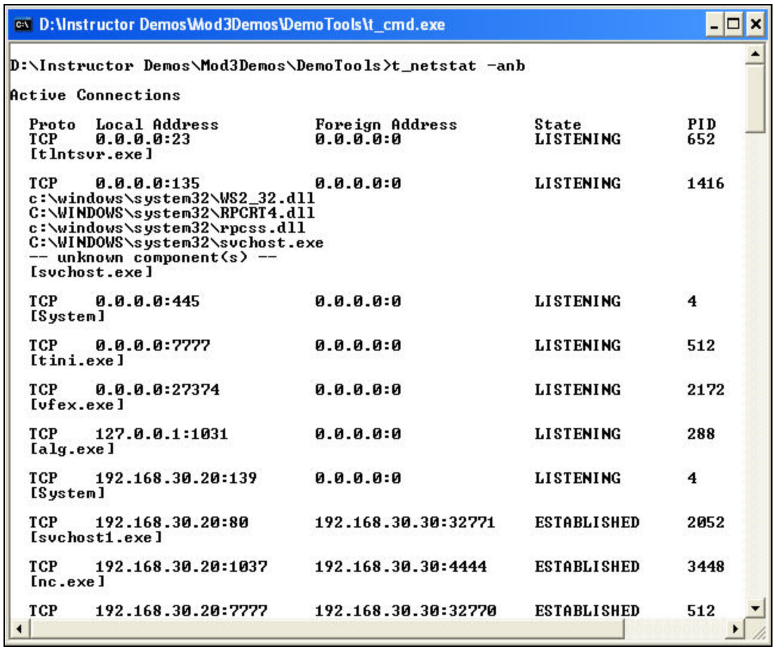


```
D:\WTools>promiscdetect.exe

PromiscDetect 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
                  - http://ntsecurity.nu/toolbox/promiscdetect/

Adapter name:

 - AMD PCNET Family PCI Ethernet Adapter

Active filter for the adapter:

 - Directed (capture packets directed to this computer)
 - Multicast (capture multicast packets for groups the computer is a member of)
 - Broadcast (capture broadcast packets)
 - Promiscuous (capture all packets on the network)

WARNING: Since this adapter is in promiscuous mode there could be a sniffer
         running on this computer!
```

Figure 63: PromiscDetect

***Netstat –anb* (Windows)**

The native *netstat –anb* command displays a list of current TCP/IP connections. It also displays the protocol of the connection, the local address (MAC address), the IP address, and the connection state. For Windows XP systems, *netstat* has an additional switch (–o) that displays the PID for the process associated with each connection.



```
D:\Instructor Demos\Mod3Demos\DemoTools>t_netstat -anb

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:23             0.0.0.0:0              LISTENING       652
  [tlntsvr.exe]

  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       1416
  c:\windows\system32\WS2_32.dll
  C:\WINDOWS\system32\RPCRT4.dll
  c:\windows\system32\rpcss.dll
  C:\WINDOWS\system32\svchost.exe
  -- unknown component(s) --
  [svchost.exe]

  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  [System]

  TCP    0.0.0.0:7777           0.0.0.0:0              LISTENING       512
  [tini.exe]

  TCP    0.0.0.0:27374          0.0.0.0:0              LISTENING       2172
  [vfex.exe]

  TCP    127.0.0.1:1031         0.0.0.0:0              LISTENING       288
  [alg.exe]

  TCP    192.168.30.20:139      0.0.0.0:0              LISTENING       4
  [System]

  TCP    192.168.30.20:80       192.168.30.30:32771    ESTABLISHED     2052
  [svchost1.exe]

  TCP    192.168.30.20:1037     192.168.30.30:4444     ESTABLISHED     3448
  [nc.exe]

  TCP    192.168.30.20:7777     192.168.30.30:32770    ESTABLISHED     512
```

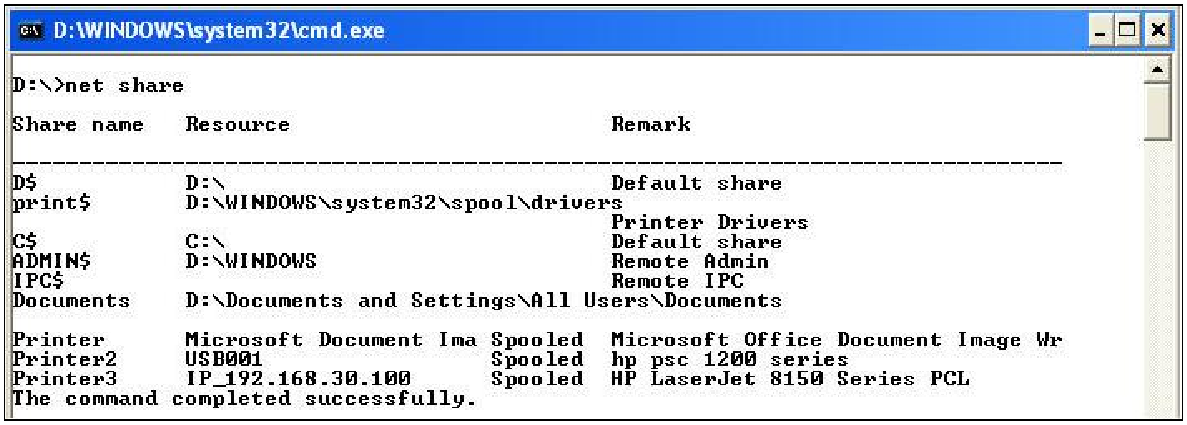
Figure 64: The netstat –anb *Command*

Nbstat (Windows)

The native *nbtstat* command can collect information about recent network connections using the NBT (NetBIOS over TCP/IP) protocol. The *nbtstat –s* command displays the session table for the local system with destination IP addresses, which helps you identify mapped connections to shared drives on other computers.

Net (Windows)

The native *net* command can be used to determine if the local machine has any network shares or is connected to any network shares. It also lists files that have been opened on the local system via a session established with a remote system over NetBios.

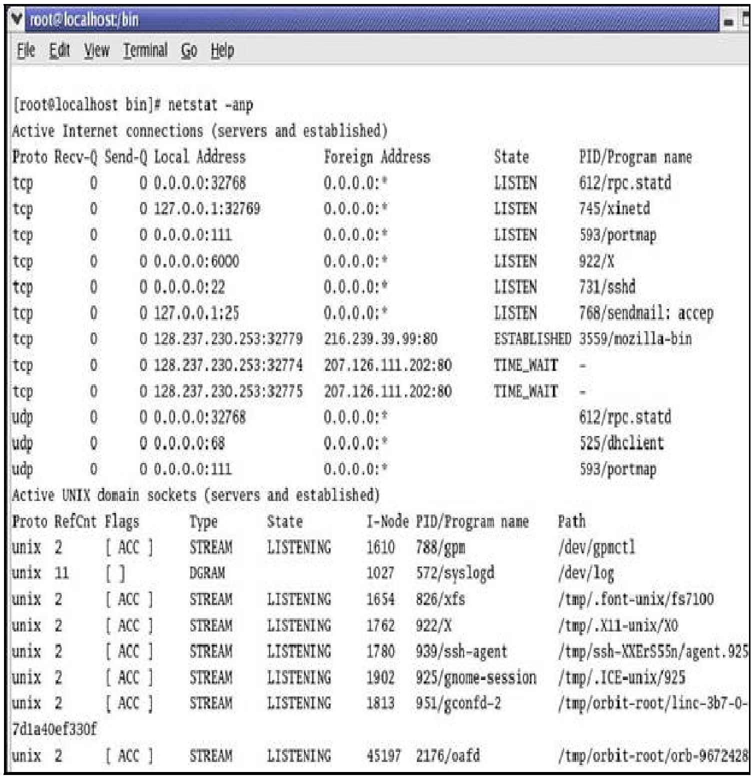


```
D:\>net share

Share name   Resource                        Remark

-------------------------------------------------------------------------------
D$           D:\                             Default share
print$       D:\WINDOWS\system32\spool\drivers
                                             Printer Drivers
C$           C:\                             Default share
ADMIN$       D:\WINDOWS                      Remote Admin
IPC$                                         Remote IPC
Documents    D:\Documents and Settings\All Users\Documents

Printer      Microsoft Document Ima Spooled  Microsoft Office Document Image Wr
Printer2     USB001                 Spooled  hp psc 1200 series
Printer3     IP_192.168.30.100      Spooled  HP LaserJet 8150 Series PCL
The command completed successfully.
```

Figure 65: The net *Command*

Netstat (Linux)

The native *netstat* command displays information on active sockets, routing tables, interfaces, masquerade connections, and multicast memberships so that you can map processes to open connections and sockets. Without parameters, *netstat* lists open sockets. With the –p parameter, it shows the process ID and name of the program that owns the socket. The *netstat –anp* command displays the protocol, local address, foreign address, state, and program responsible for the open connection.

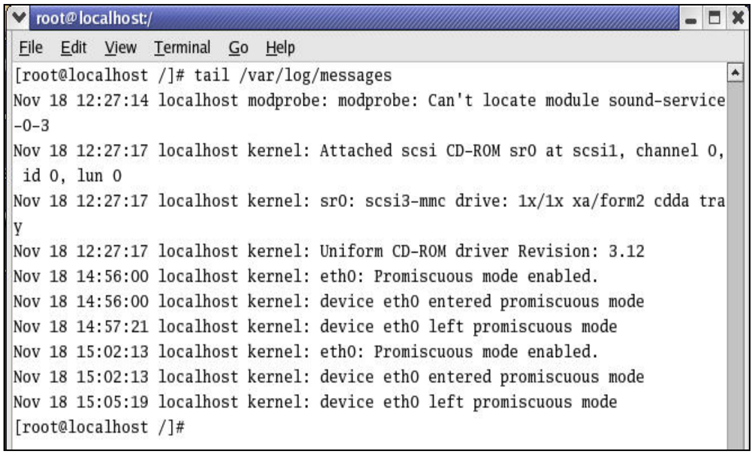


```
[root@localhost bin]# netstat -anp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:32768           0.0.0.0:*               LISTEN      612/rpc.statd
tcp        0      0 127.0.0.1:32769         0.0.0.0:*               LISTEN      745/xinetd
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      593/portmap
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN      922/X
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      731/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      768/sendmail: accep
tcp        0      0 128.237.230.253:32779   216.239.39.99:80        ESTABLISHED 3559/mozilla-bin
tcp        0      0 128.237.230.253:32774   207.126.111.202:80      TIME_WAIT   -
tcp        0      0 128.237.230.253:32775   207.126.111.202:80      TIME_WAIT   -
udp        0      0 0.0.0.0:32768           0.0.0.0:*                           612/rpc.statd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           525/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*                           593/portmap
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node PID/Program name    Path
unix  2      [ ACC ]     STREAM     LISTENING     1610   788/gpm             /dev/gpmctl
unix  11     [ ]         DGRAM                    1027   572/syslogd         /dev/log
unix  2      [ ACC ]     STREAM     LISTENING     1654   826/xfs             /tmp/.font-unix/fs7100
unix  2      [ ACC ]     STREAM     LISTENING     1762   922/X               /tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     1780   939/ssh-agent       /tmp/ssh-XXErS55n/agent.925
unix  2      [ ACC ]     STREAM     LISTENING     1902   925/gnome-session   /tmp/.ICE-unix/925
unix  2      [ ACC ]     STREAM     LISTENING     1813   951/gconfd-2        /tmp/orbit-root/linc-3b7-0-
7d1a40ef330f
unix  2      [ ACC ]     STREAM     LISTENING     45197  2176/oafd           /tmp/orbit-root/orb-9672428
```

Figure 66: The netstat –anp *Command*

/var/log/messages (Linux)

Look at */var/log/messages* to determine if the network interface card is in promiscuous mode.



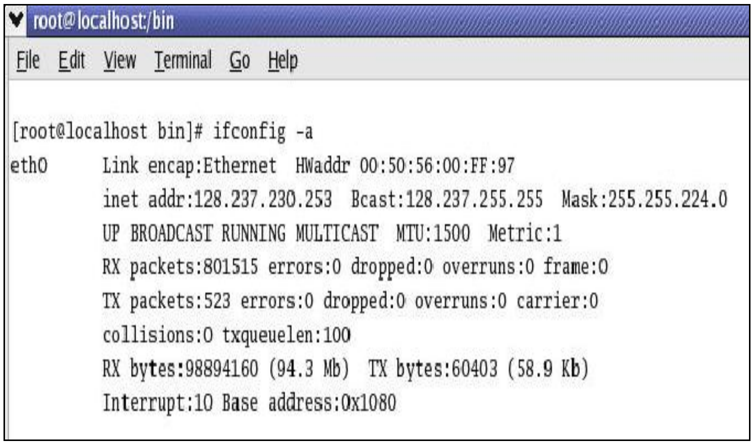
```
[root@localhost /]# tail /var/log/messages
Nov 18 12:27:14 localhost modprobe: modprobe: Can't locate module sound-service
-0-3
Nov 18 12:27:17 localhost kernel: Attached scsi CD-ROM sr0 at scsi1, channel 0,
 id 0, lun 0
Nov 18 12:27:17 localhost kernel: sr0: scsi3-mmc drive: 1x/1x xa/form2 cdda tra
y
Nov 18 12:27:17 localhost kernel: Uniform CD-ROM driver Revision: 3.12
Nov 18 14:56:00 localhost kernel: eth0: Promiscuous mode enabled.
Nov 18 14:56:00 localhost kernel: device eth0 entered promiscuous mode
Nov 18 14:57:21 localhost kernel: device eth0 left promiscuous mode
Nov 18 15:02:13 localhost kernel: eth0: Promiscuous mode enabled.
Nov 18 15:02:13 localhost kernel: device eth0 entered promiscuous mode
Nov 18 15:05:19 localhost kernel: device eth0 left promiscuous mode
[root@localhost /]#
```

Figure 67: /var/log/messages

Ifconfig (Linux)

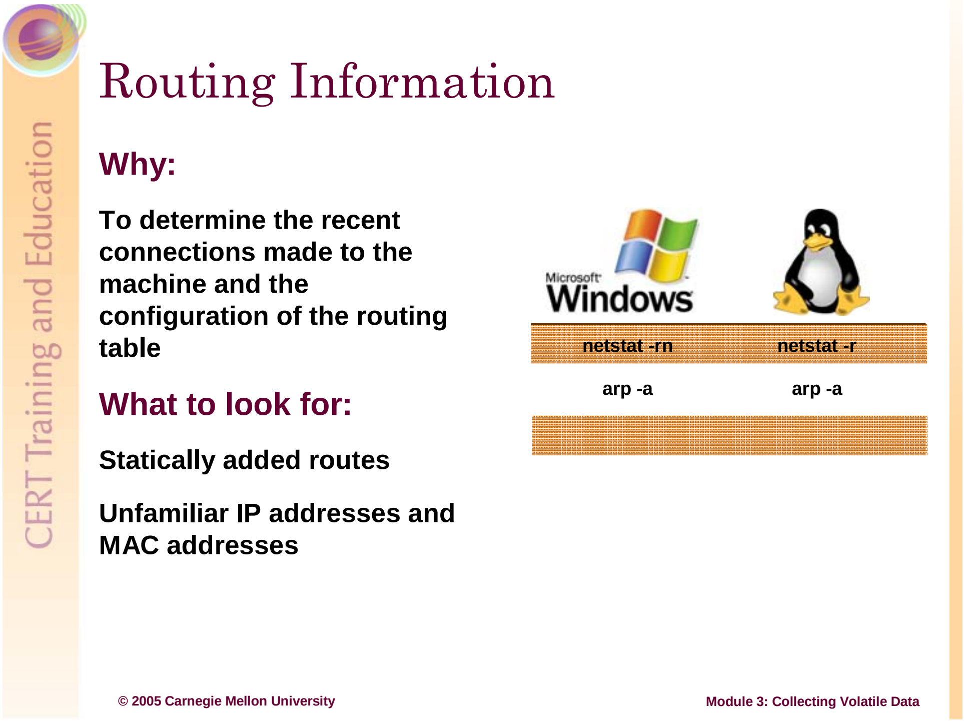
The native *ifconfig* command displays the current configuration for a network interface.

Displayed information includes IP address, gateway, DNS servers, and promiscuous mode detection. In Figure 68, there would be a Promisc flag next to Multicast to alert you that there is a network sniffer on this machine.



```
[root@localhost bin]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:50:56:00:FF:97
          inet addr:128.237.230.253  Bcast:128.237.255.255  Mask:255.255.224.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:801515 errors:0 dropped:0 overruns:0 frame:0
          TX packets:523 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:98894160 (94.3 Mb)  TX bytes:60403 (58.9 Kb)
          Interrupt:10 Base address:0x1080
```

Figure 68: The ifconfig *Command*



CERT Training and Education

Routing Information

Why:

To determine the recent connections made to the machine and the configuration of the routing table

What to look for:

Statically added routes

Unfamiliar IP addresses and MAC addresses

Microsoft Windows

netstat -rn	netstat -r
arp -a	arp -a

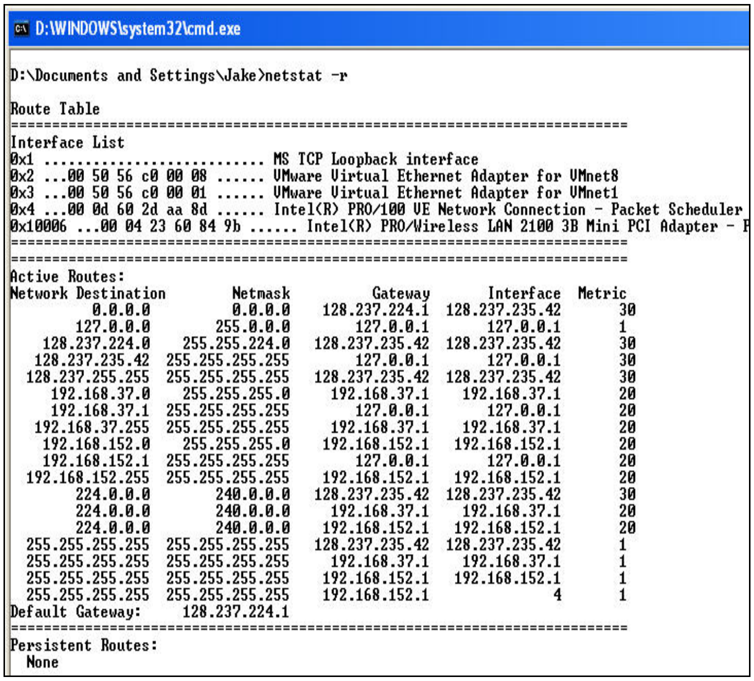
Module 3: Collecting Volatile Data

3.9.2.2 Routing Information

As you use the tools in this section to collect routing information, be sure to pay attention to the suspicious computer’s routing table. Make note of added or unfamiliar routes. Also, look at the ARP cache to identify recent connections.

Netstat, Route (Windows)

The native *netstat* and *route* commands allow you to collect routing information. As shown in Figure 69, *netstat –r* displays all volatile active routes for the suspicious computer.

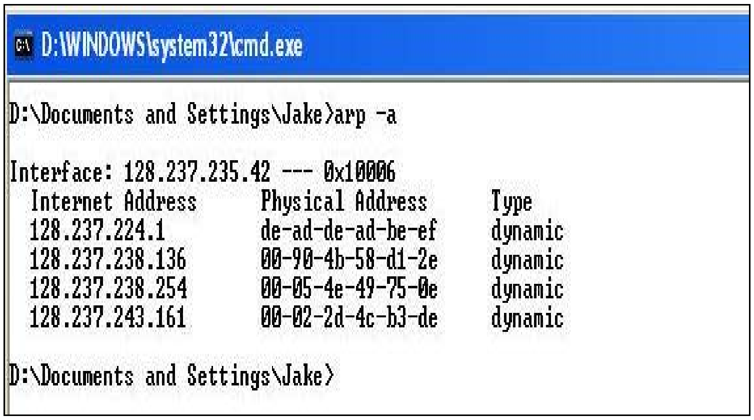


```
D:\Documents and Settings\Jake>netstat -r

Route Table
============================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 50 56 c0 00 08 ...... VMware Virtual Ethernet Adapter for VMnet8
0x3 ...00 50 56 c0 00 01 ...... VMware Virtual Ethernet Adapter for VMnet1
0x4 ...00 0d 60 2d aa 8d ...... Intel(R) PRO/100 VE Network Connection - Packet Scheduler
0x10006 ...00 04 23 60 84 9b ...... Intel(R) PRO/Wireless LAN 2100 3B Mini PCI Adapter - P
============================================================================
============================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    128.237.224.1   128.237.235.42      30
        127.0.0.0        255.0.0.0        127.0.0.1        127.0.0.1       1
    128.237.224.0    255.255.224.0   128.237.235.42   128.237.235.42      30
   128.237.235.42  255.255.255.255        127.0.0.1        127.0.0.1      30
  128.237.255.255  255.255.255.255   128.237.235.42   128.237.235.42      30
     192.168.37.0    255.255.255.0     192.168.37.1     192.168.37.1      20
     192.168.37.1  255.255.255.255        127.0.0.1        127.0.0.1      20
   192.168.37.255  255.255.255.255     192.168.37.1     192.168.37.1      20
    192.168.152.0    255.255.255.0    192.168.152.1    192.168.152.1      20
    192.168.152.1  255.255.255.255        127.0.0.1        127.0.0.1      20
  192.168.152.255  255.255.255.255    192.168.152.1    192.168.152.1      20
        224.0.0.0        240.0.0.0   128.237.235.42   128.237.235.42      30
        224.0.0.0        240.0.0.0     192.168.37.1     192.168.37.1      20
        224.0.0.0        240.0.0.0    192.168.152.1    192.168.152.1      20
  255.255.255.255  255.255.255.255   128.237.235.42   128.237.235.42       1
  255.255.255.255  255.255.255.255     192.168.37.1     192.168.37.1       1
  255.255.255.255  255.255.255.255    192.168.152.1    192.168.152.1       1
  255.255.255.255  255.255.255.255    192.168.152.1                4       1
Default Gateway:     128.237.224.1
============================================================================
Persistent Routes:
  None
```

Figure 69: The Windows netstat –r *Command*

Arp (Windows)

The native *arp* command displays the following information about the suspicious computer's network interface: IP address, MAC address, and type (dynamic or static). The ARP cache also stores the MAC address to IP address translations for the last two minutes [Pierce 03]. The –a switch pulls the active ARP cache entries.

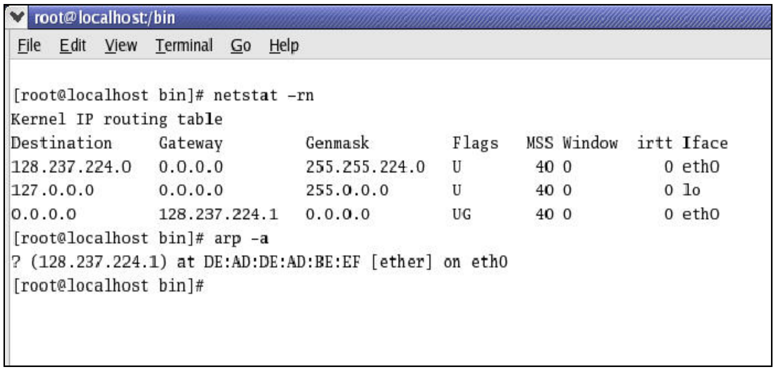


```
D:\Documents and Settings\Jake>arp -a

Interface: 128.237.235.42 --- 0x10006
  Internet Address      Physical Address      Type
  128.237.224.1         de-ad-de-ad-be-ef     dynamic
  128.237.238.136       00-90-4b-58-d1-2e     dynamic
  128.237.238.254       00-05-4e-49-75-0e     dynamic
  128.237.243.161       00-02-2d-4c-b3-de     dynamic

D:\Documents and Settings\Jake>
```

Figure 70: The Windows arp *Command*

Netstat (Linux)

The native *netstat –rn* command displays the current routing table and gateways for all routes on the suspicious computer.

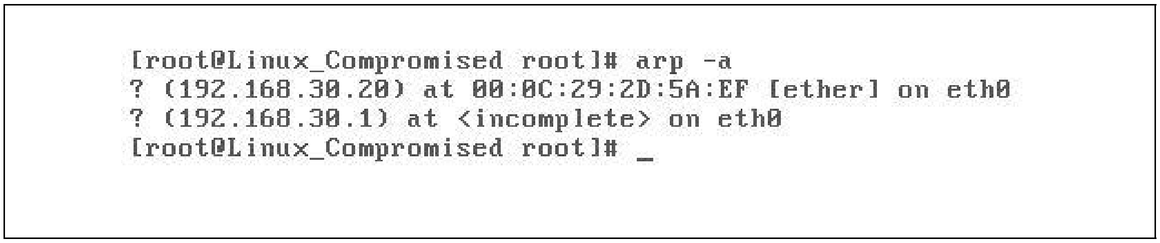


```
[root@localhost bin]# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
128.237.224.0   0.0.0.0         255.255.224.0   U        40 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U        40 0          0 lo
0.0.0.0         128.237.224.1   0.0.0.0         UG       40 0          0 eth0
[root@localhost bin]# arp -a
? (128.237.224.1) at DE:AD:DE:AD:BE:EF [ether] on eth0
[root@localhost bin]#
```

Figure 71: The Linux netstat –rn *Command*

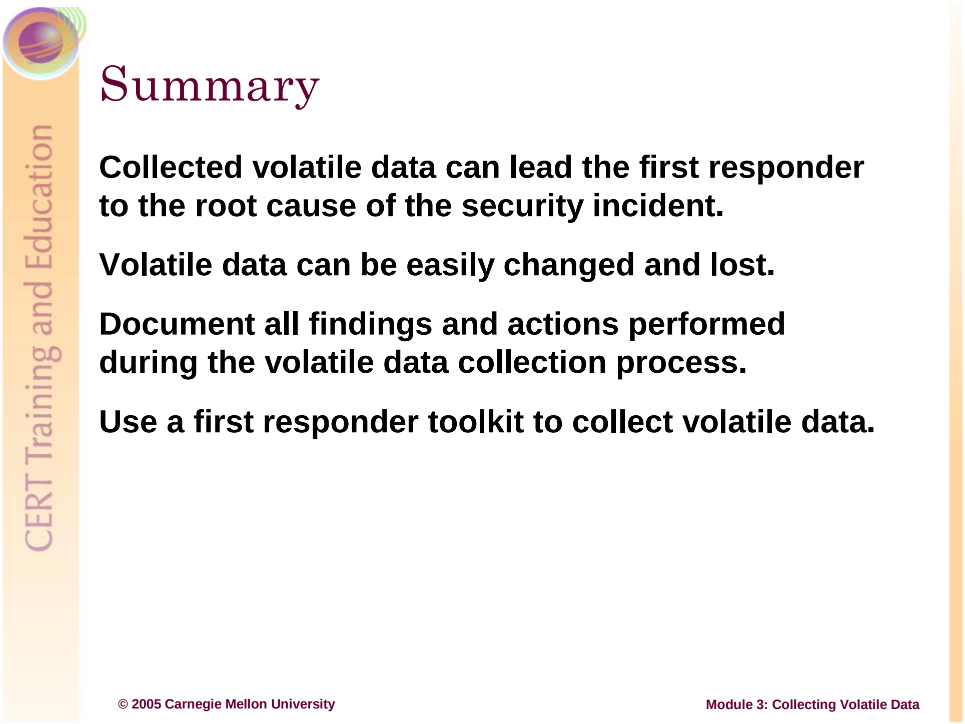
Arp (Linux)

The native *arp –a* command displays route entries for the suspicious computer. Determine whether there are any permanent ARP cache entries or whether ARP proxies were created.



```
[root@Linux_Compromised root]# arp -a
? (192.168.30.20) at 00:0C:29:2D:5A:EF [ether] on eth0
? (192.168.30.1) at <incomplete> on eth0
[root@Linux_Compromised root]# _
```

Figure 72: The Linux arp –a *Command*



Summary

Collected volatile data can lead the first responder to the root cause of the security incident.

Volatile data can be easily changed and lost.

Document all findings and actions performed during the volatile data collection process.

Use a first responder toolkit to collect volatile data.

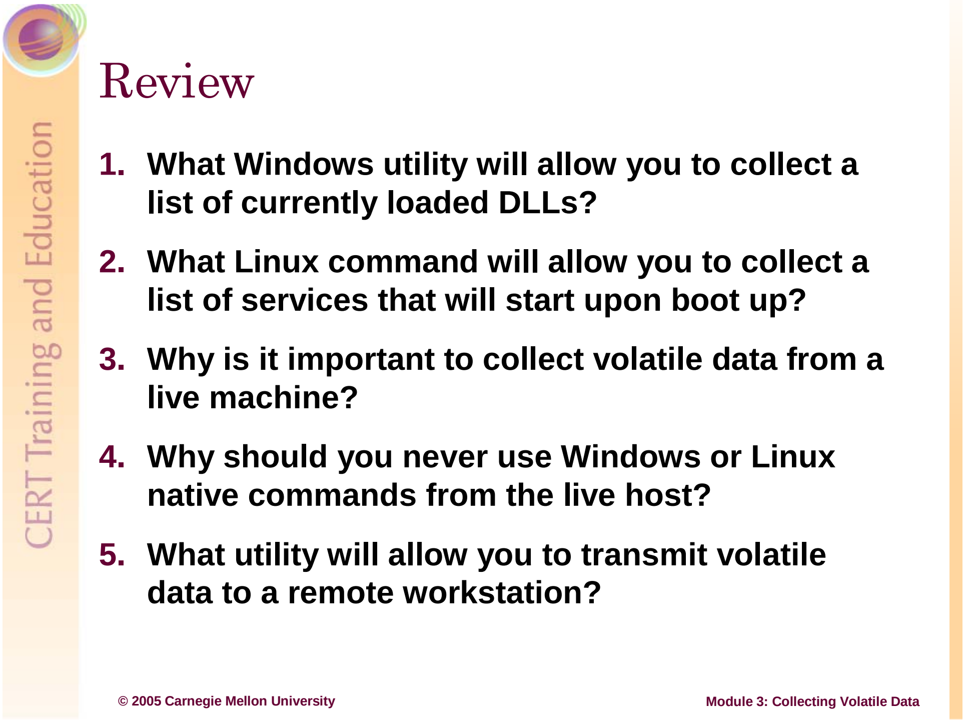
CERT Training and Education

© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.10 Summary

The slide above summarizes the main points for Module 3.



CERT Training and Education

Review

1. What Windows utility will allow you to collect a list of currently loaded DLLs?
2. What Linux command will allow you to collect a list of services that will start upon boot up?
3. Why is it important to collect volatile data from a live machine?
4. Why should you never use Windows or Linux native commands from the live host?
5. What utility will allow you to transmit volatile data to a remote workstation?

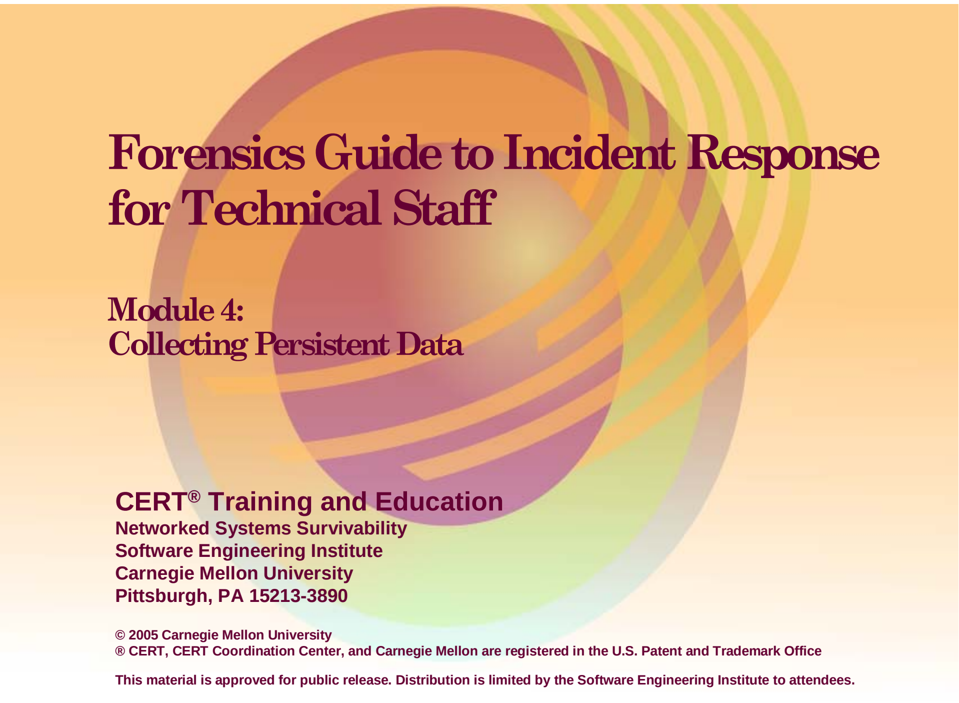
© 2005 Carnegie Mellon University

Module 3: Collecting Volatile Data

3.11 Review

1. The *Listdlls.exe* utility will allow you to collect a list of currently loaded DLLs.
2. The *chkconfig* will allow you to collect a list of services that will start upon boot up.
3. It is important to collect volatile data from a live machine for the following reasons:
    a. to gain initial insight
    b. to determine a logical timeline of the computer security incident
    c. to determine the next course of action
    d. because you have one chance to collect volatile data
4. You should never use native Windows or Linux commands on the suspect live host because the integrity of the suspect machine could be compromised.
5. The *Netcat* or *Cryptcat* utilities will allow you to transmit volatile data to a remote workstation.

4 Module 4: Collecting Persistent Data



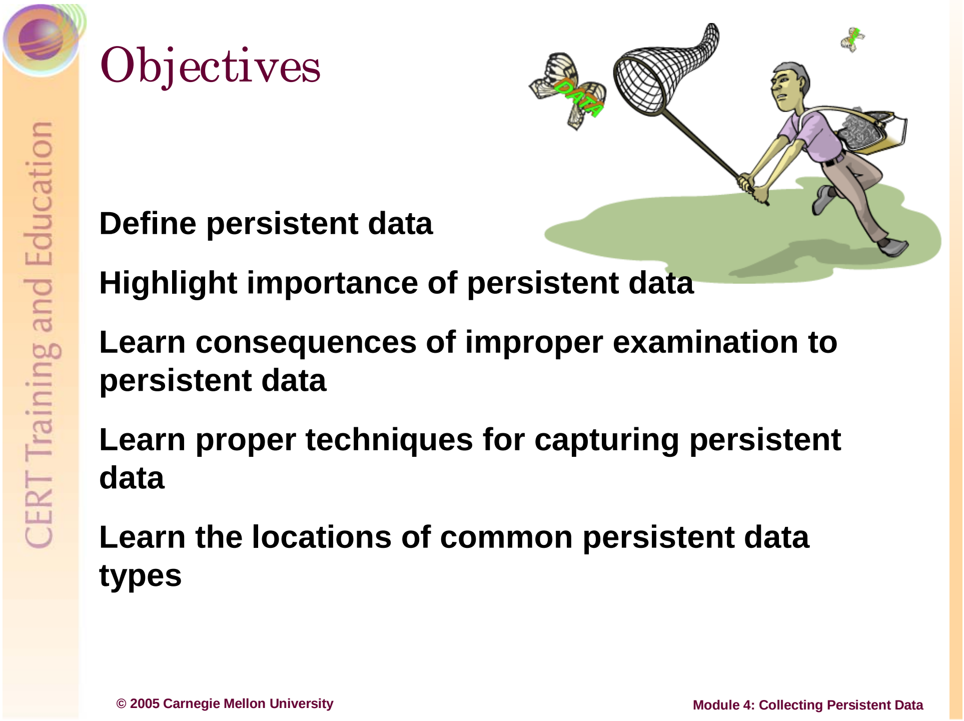
Forensics Guide to Incident Response for Technical Staff

Module 4:
Collecting Persistent Data

CERT® Training and Education
Networked Systems Survivability
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Most epistemologists concur that for effective discussion to take place, a group must first agree on what they are talking about. Therefore, we define *persistent data* as that which remains unchanged after the computer has been powered off. This data plays an important part in any type of computer event investigation. While not as effervescent as volatile data, persistent data can still be contaminated through improper handling.

This module reviews techniques for capturing persistent data in a forensically sound manner. It also covers the location of common persistent data types.



CERT Training and Education

Objectives

Define persistent data

Highlight importance of persistent data

Learn consequences of improper examination to persistent data

Learn proper techniques for capturing persistent data

Learn the locations of common persistent data types

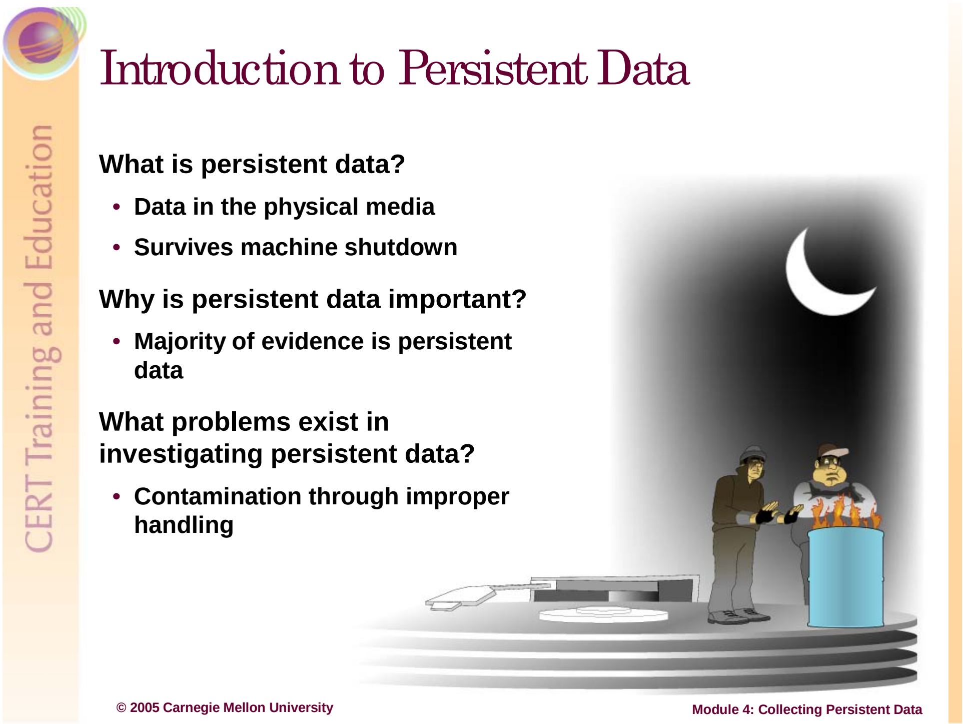
© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.1 Objectives

The primary objective of this module is to highlight the importance of handling persistent data in a forensically sound manner. It includes the following topics:

- definition of persistent data
- importance of persistent data in responding to a security event
- consequences of improper examination
- proper techniques for capturing persistent data
- locations of common persistent data types



CERT Training and Education

Introduction to Persistent Data

What is persistent data?

- Data in the physical media
- Survives machine shutdown

Why is persistent data important?

- Majority of evidence is persistent data

What problems exist in investigating persistent data?

- Contamination through improper handling

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.2 Introduction to Persistent Data

4.2.1 What Is Persistent Data?

Fundamentally, persistent data is retained and remains unchanged after the computer has been powered off. It is found on removable storage media as well as hard drives. Despite its stable nature, precautions still need to be taken to ensure proper collection.

This definition is broad enough to include data held on disks, flash memory, cell phones, and PDAs. While the details may differ, the underlying philosophy of handling persistent data is the same across the different media.

4.2.2 Why is Persistent Data Important?

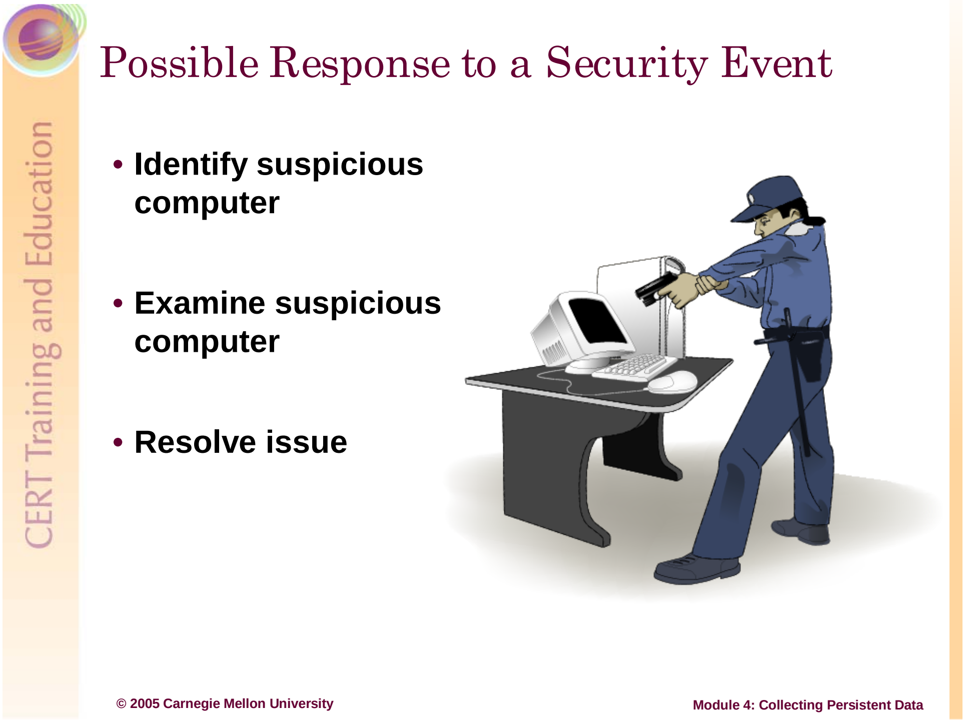
Most forensics evidence is found in persistent data. Examples include documents, emails, Web activity, and deleted files.

4.2.3 What Problems Exist in Investigating Persistent Data?

Investigating persistent data is not usually complicated. With access to a computer, most users can search for files, investigate past Internet activity, or even check for the presence of inappropriate programs. However, exploring the suspicious computer in this manner will alter evidence and make it far less useful.

Applying the needle/haystack metaphor—the best way to get the needle, if it is there, is to collect the entire haystack—highlights two issues:

1. It can take a long time to collect a haystack. Depending on the situation, it may be better to collect only those areas of the hard drive that have the highest probability of containing the information you seek. This is determined by the indicators prompting the investigation (i.e., inappropriate Web use, harassing emails, or malicious software).
2. If the collection process takes a long time, it can take even longer to comb through the haystack for the needle.



CERT Training and Education

Possible Response to a Security Event

- Identify suspicious computer
- Examine suspicious computer
- Resolve issue

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

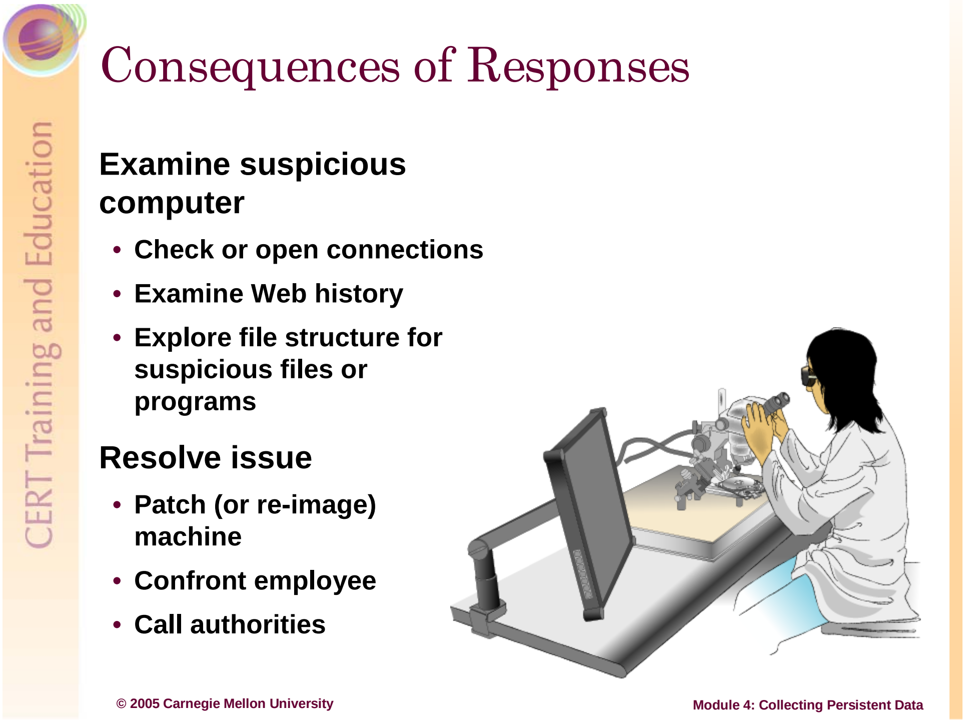
4.3 Responding to a Security Event

The following steps are possible responses to a security event. Unfortunately, they may not be the best response.

- Identify the suspicious computer.
  - IDS alert
  - user complaint of poor/odd computer performance
  - user suspected of inappropriate computer use
- Examine the suspicious computer.
  - examine Web history
  - check or open connections
  - explore file structure for suspicious files or programs
- Resolve the issue.
  - patch (or re-image) machine
  - confront employee
  - call authorities

Once a computer comes to the attention of a system administrator, there are several possible responses to a security event, and some are better than others. It is possible to discover useful information by directly accessing the suspicious computer. By looking in the right places, an inquisitive user can, with a fair degree of accuracy, determine the user’s Internet activity, examine emails, check for inappropriate material, and dig up (or at least discover the existence of) deleted files.

However, there maybe legal consequences to satisfying one’s curiosity in this manner.



CERT Training and Education

Consequences of Responses

Examine suspicious computer

- **Check or open connections**
- **Examine Web history**
- **Explore file structure for suspicious files or programs**

Resolve issue

- **Patch (or re-image) machine**
- **Confront employee**
- **Call authorities**

Module 4: Collecting Persistent Data

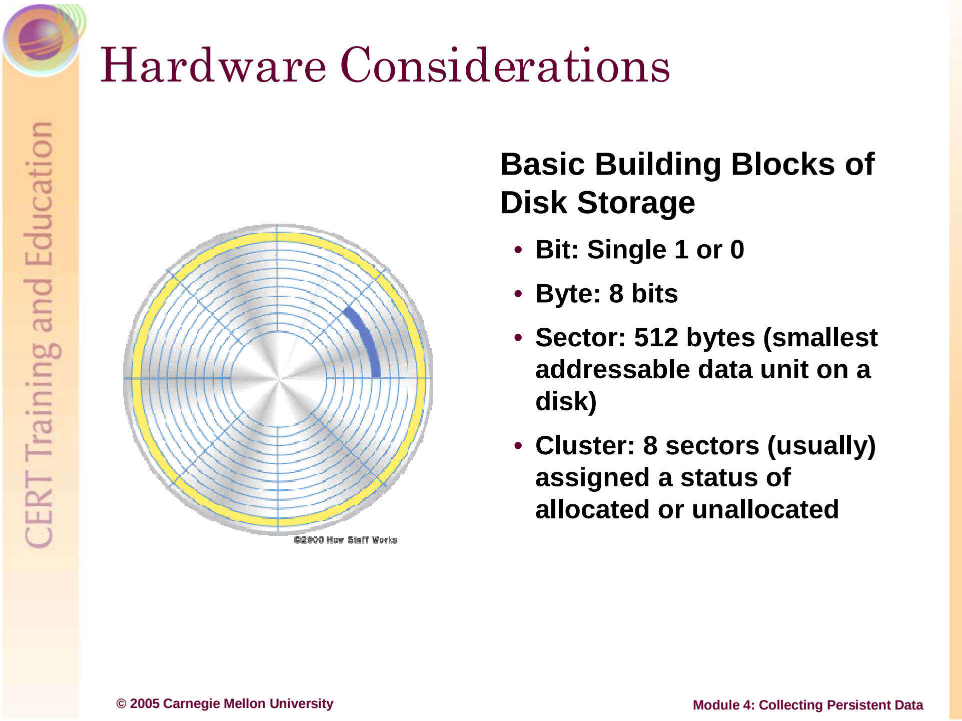
4.3.1 Consequences of Responses

An examination of the suspicious computer, while satisfying curiosity, will wreak havoc on the future value of the information gathered. The computer should be thought of as a crime scene. Consider this scenario: an apartment has been burglarized. The tenant calls the landlord, who comes in, opens drawers and closets, and looks behind pictures and under furniture. He finds a wallet, goes through its contents, and discovers that the owner is a tenant two floors up.

From one perspective, the landlord has just solved the case with a minimum of fuss. He knows the identity of the burglar and can confront the perpetrator and attempt to resolve the situation. Once the perpetrator is confronted, his conscience may be pricked so that he is motivated to return any stolen property and promises not to break into any more apartments. After all, he was only testing the apartment's integrity. He wasn't really stealing anything; just taking temporary custody of certain objects to highlight potential security concerns.

From another perspective, the landlord's enthusiasm to solve the problem really complicated the situation. By contaminating the crime scene and handling the evidence, the landlord has made it difficult for forensic examiners to develop a case against the thief that could be successfully prosecuted. By confronting the thief, the landlord has warned him that his identity and illicit activities have been discovered. The thief may now take additional measures during his next burglary to remain undetected.

While the analogy is not perfect, it does illustrate some of the problems facing system administrators when responding to a security event. There will be times when it is clear that a forensic examiner needs to be called in. There will also be times when it is clear that the problem can be easily fixed and forgotten. This module is for all the times in between. By jumping straight into a suspicious computer, timestamps are changed and evidence in the slack space or unallocated space may be overwritten. If data is not collected in a forensically sound manner, defense attorneys can claim that the incriminating data was planted.



CERT Training and Education

Hardware Considerations

Basic Building Blocks of Disk Storage

- Bit: Single 1 or 0
- Byte: 8 bits
- Sector: 512 bytes (smallest addressable data unit on a disk)
- Cluster: 8 sectors (usually) assigned a status of allocated or unallocated

©2000 How Stuff Works

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.4 Basic Building Blocks of Disk Storage

This is a quick review of hardware information covered in the Module 2. Understanding how memory is physically organized helps in understanding how the computer stores data and how that data can linger even after it has been deleted.

Bit: A single 1 or 0

Byte: 8 bits

Sector: 512 bytes (smallest addressable data unit on a disk)

Cluster: 8 sectors (usually) assigned a status of allocated or unallocated

Figure 73 illustrates sectors and clusters on a disk.[14]

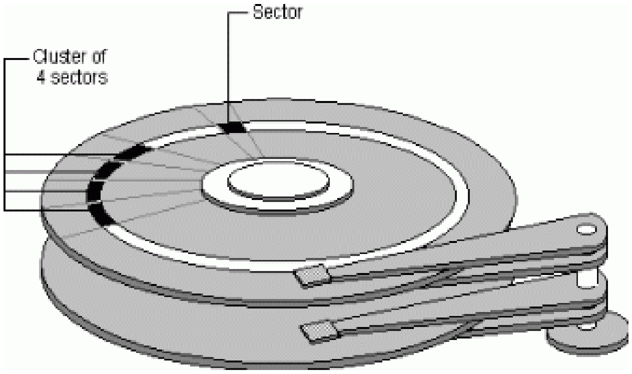
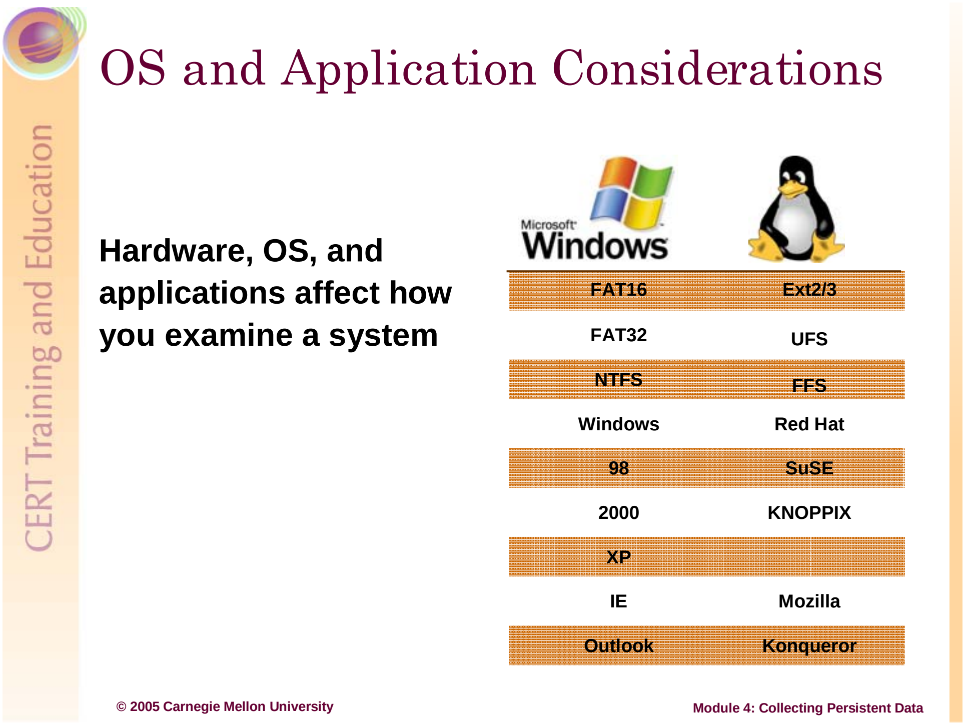


Figure 73: Sectors and Clusters

[14] *http://www.microsoft.com/technet/archive/winntas/support/diskover.mspx*



CERT Training and Education

OS and Application Considerations

Hardware, OS, and applications affect how you examine a system

FAT16	Ext2/3
FAT32	UFS
NTFS	FFS
Windows	Red Hat
98	SuSE
2000	KNOPPIX
XP	
IE	Mozilla
Outlook	Konqueror

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.5 OS and Application Considerations

4.5.1 Windows

4.5.1.1 FAT

One of the earlier file systems, the file allocation table (FAT) comes in three variations: FAT12, FAT16, and FAT32. FAT 32 is the newest variation included in Windows 95 release 2 and later [Honeynet 04]. The FAT includes an entry for each cluster on the disk except 0 and 1. Each entry includes pointers to the clusters that make up a file. The number indicates entry size (i.e., FAT32 has a 32-bit table entry).

4.5.1.2 NTFS

The new technology file system (NTFS) uses a master file table (MFT) to keep track of information on a disk. The MFT provides greater functionality and allows for larger file sizes than the FAT.

In NTFS, a file can be stored two ways. If the file is small enough, it can be stored in the MFT itself and is called a *resident file*. If the file is larger, it is stored in assigned clusters and called a *non-resident file*.

Non-resident files' names and locations are stored in the MFT. They may span several clusters, but a cluster can only be allocated to one file at a time. While clusters can be reused after a file is deleted, two different files can never share a cluster.

4.5.2 Linux/UNIX

4.5.2.1 Ext2/3

Ext2/3 is the predominant file system for Linux. Ext2 partitions are organized into groups, which are made up of inodes and blocks. The inode tables contain information on files located in each group. Reiser is one flavor of the Ext2 file system. Ext3 allows for journaling in the ext file system.

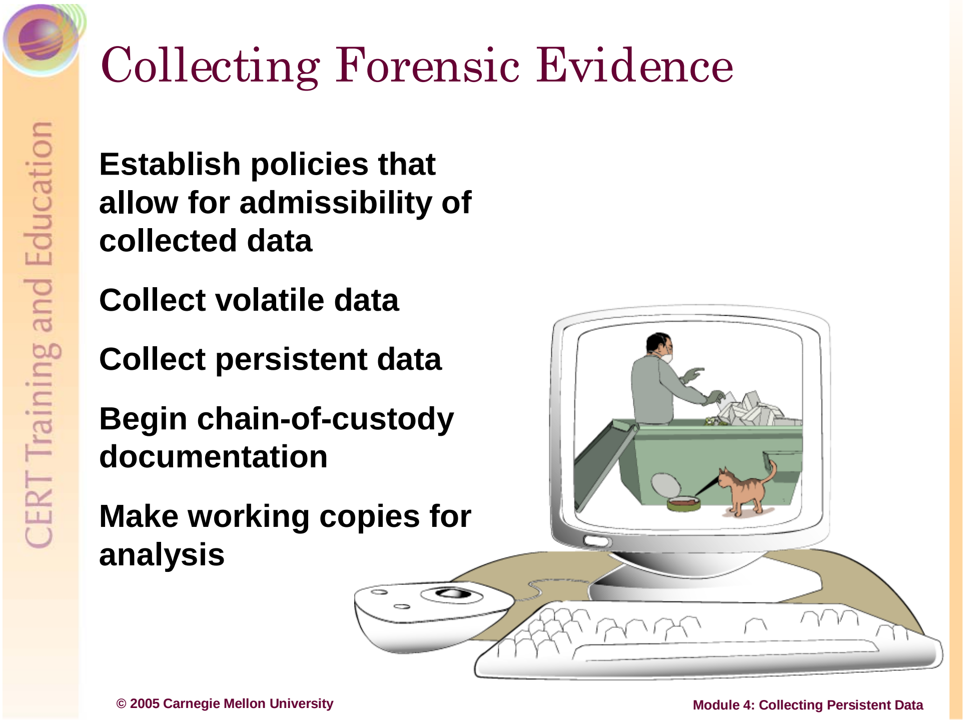
Linux and UNIX developed from common ancestry and many flavors exist today. Some of the more popular ones are listed below. For more information, visit http://encyclopedia.thefreedictionary.com/UFS.

- UFS (UNIX File System)
  A common UNIX file system
- FFS (Fast File System)
  A common file system for FreeBSD, Open BSD, and NetBSD
- HFS and HFS+ (Hierarchical File System)
  Common file systems for Macintoshes

4.5.3 Operating Systems

Each OS and application may organize information its own way. The Windows OS, for example, generally has Microsoft's Internet Explorer (IE) and Outlook. IE always keeps the cookie file in the same place (the *\Documents and Settings\[user]\Cookies* folder). The uniformity of Windows makes it easy to find certain types of information.

Linux is more complicated; Linux flavors may organize information differently, and builds within the same flavor may even differ from each other. Because Linux systems may be highly customized, certain types of information are not always stored in the same place. For this reason, you should have a good understanding of where critical files are located for all OSs on your network before a security event occurs.



CERT Training and Education

Collecting Forensic Evidence

Establish policies that allow for admissibility of collected data

Collect volatile data

Collect persistent data

Begin chain-of-custody documentation

Make working copies for analysis

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.6 Collecting Forensic Evidence

Following an established procedure has benefits. As you create the procedure, potential issues can be spotted and corrected before real problems occur. As you use the procedure, steps are done in a consistent, repeatable manner. Below is a sample, high-level collection procedure:

1. Establish policies that allow for admissibility of collected data.

   This is particularly important in light of admissibility of computer records. In order for a computer record to receive an exception to the heresy rules, it must be collected and be used in the normal course of business. If you plan on using any logs in your investigation and subsequent prosecution, those logs should be a part of your normal business process and not just a one-time effort.

2. Document reasons for beginning the examination.

   This could be anything from poor system performance to indications that company policy has been violated. The important issue it that there is a documented reason for the investigation.

3. Document the scene.

   It can be tempting to get lost in the virtual environments of networks and hard drives when responding to a security incident. One must not, however, overlook clues in the physical world. The best method to document a physical scene is to use a digital camera. Take pictures of the computer. Focus especially on the peripherals and network connections. Describe in the log what was connected to the computer as well as its environment. Was it in the middle of a busy cubicle farm or tucked away in a secluded corner? These details may be important to a forensic examiner.

4. Collect volatile data.

   As described in Module 3, volatile data is effervescent and should be collected before nonvolatile data.

5. Collect persistent data and document each step.

   a. Calculate hash values of the suspicious computer’s disk/files.

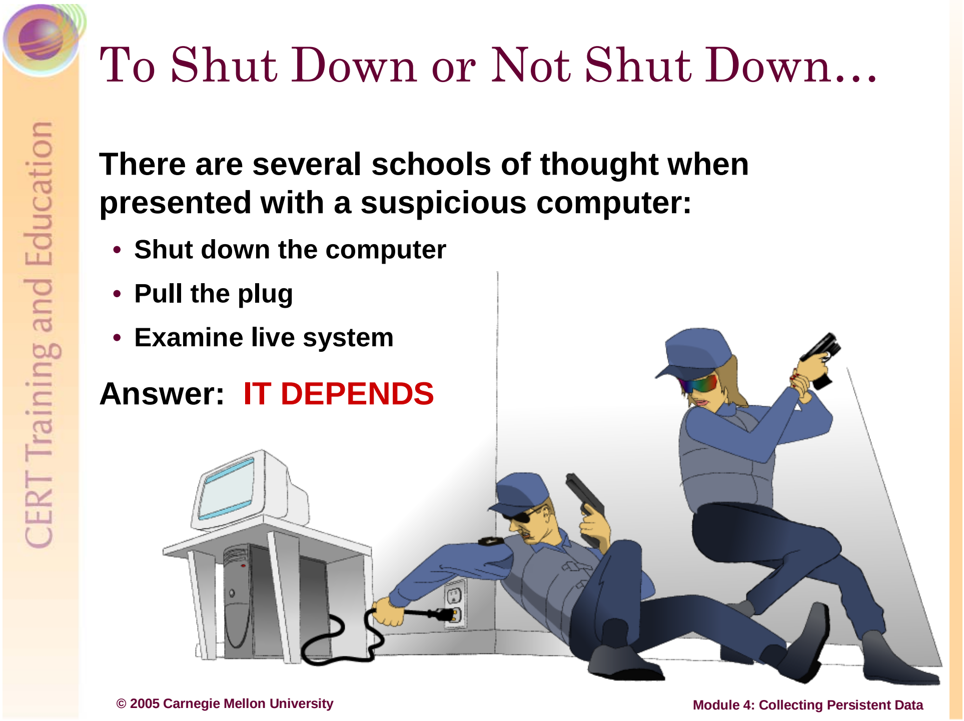
   b. Perform a bit-stream duplicate image/copy.

   c. Calculate the hash values of the newly created master disk/file image to verify authenticity.

6. Begin chain-of-custody documentation (original evidence and master copy) and document each step.

   a. Make working copies for analysis and verify their authenticity by calculating hash values.

   b. Begin initial analysis to determine if additional resources are necessary.



CERT Training and Education

To Shut Down or Not Shut Down…

There are several schools of thought when presented with a suspicious computer:

- **Shut down the computer**
- **Pull the plug**
- **Examine live system**

Answer: IT DEPENDS

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.6.1 To Shut Down or Not to Shut Down…

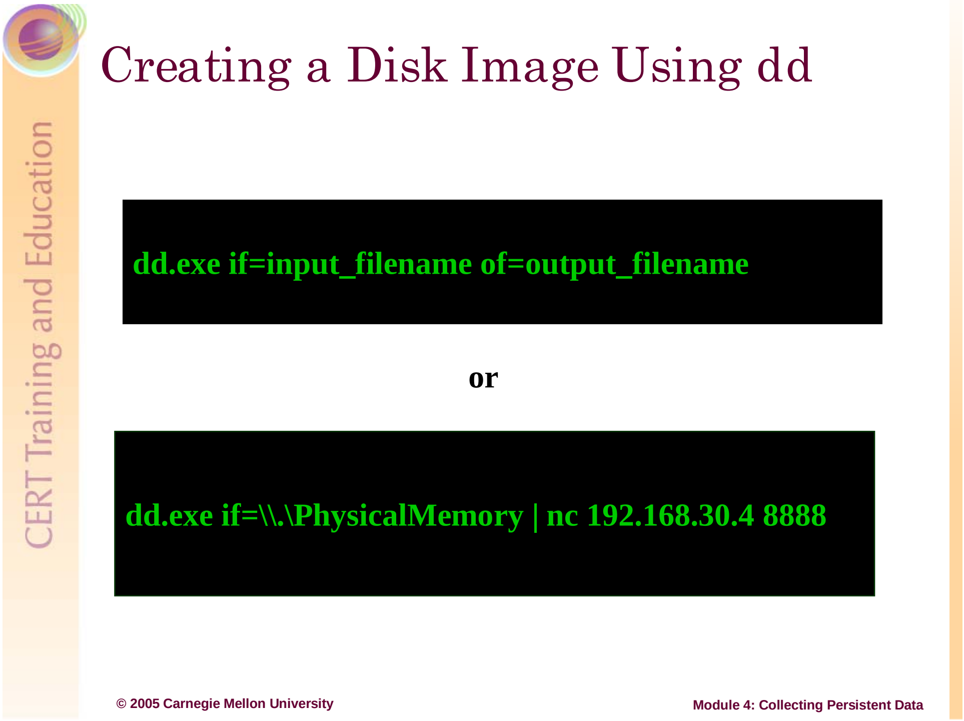
It is accepted that the action of switching off the computer may mean that a small amount of evidence may be unrecoverable if it has not been saved to the memory but the integrity of the evidence already present will be retained [ACPO 03].

Volatile data must be collected from a live system. Collecting nonvolatile data is a different story. Shutting down a system may trigger malicious software (malware) to erase evidence. If you suspect that malware is present, just pull the plug. Understanding the principles behind nonvolatile data collection will assist those responding to a security event to make the decision that best fits the circumstances of the security incident.

After a computer has been turned off, there are two options:

1. Remove the hard drive and mount it read-only on a forensic examination computer.
2. If you can not remove the hard drive, boot the computer from a safe boot disk, like Knoppix STD or Helix. Both are loaded with the necessary forensic tools to image the suspicious drive and send it across the network to another location.

Some computers may be running critical processes and cannot be shut down. In a case like this, it is vital to have a response CD as described in Module 2.



CERT Training and Education

Creating a Disk Image Using dd

```
dd.exe if=input_filename of=output_filename
```

or

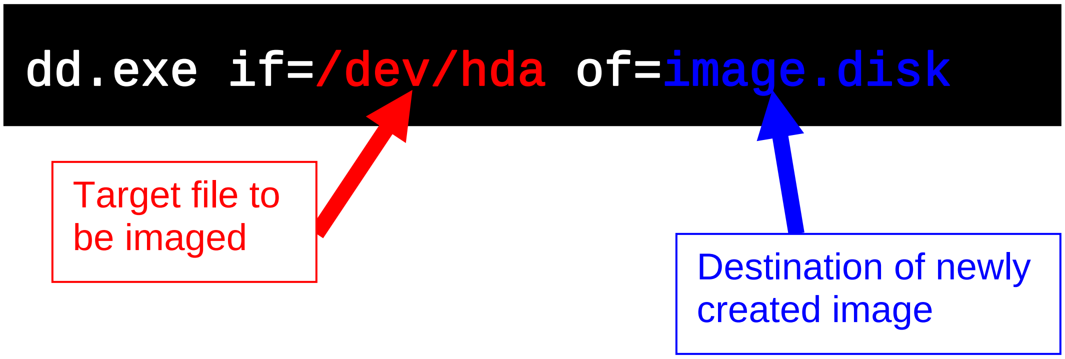
```
dd.exe if=\\.\PhysicalMemory | nc 192.168.30.4 8888
```

Module 4: Collecting Persistent Data

4.6.2 Creating a Disk Image Using dd

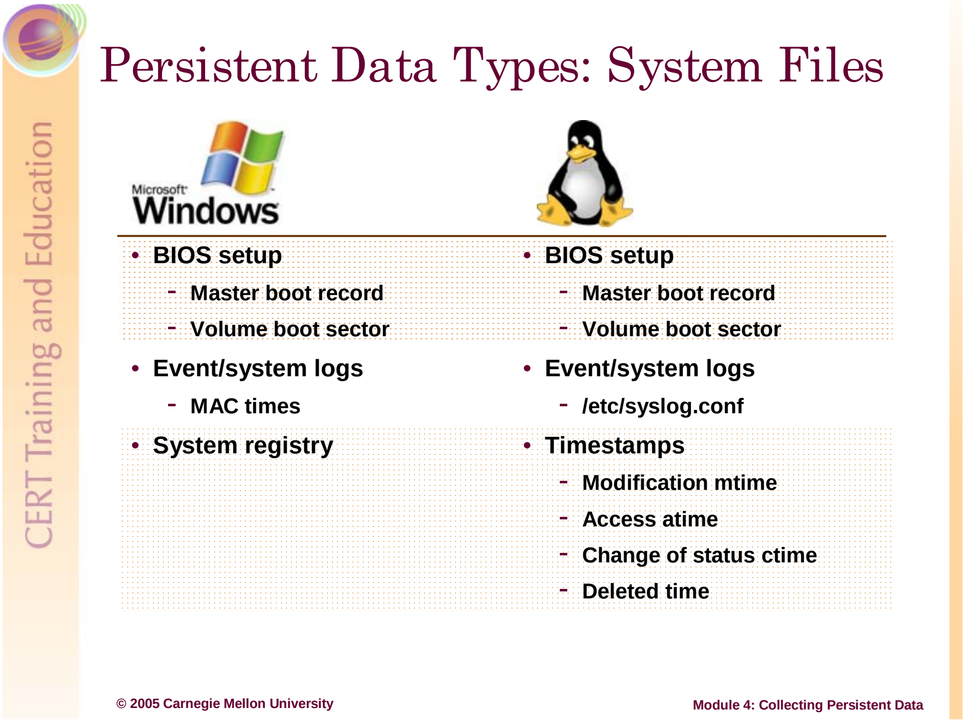
There are several tools that can image a disk or partition. We will use dd.exe, a small but powerful tool available for Windows and Linux. Following are some basic commands for dd [Grundy 04]. For Linux systems, most versions have dd as a native tool. For Windows systems, download dd from http://uranus.it.swin.edu.au/~jn/linux/rawwrite/dd.htm.

The basic syntax for dd.exe is explained in Figure 74.



```
dd.exe if=/dev/hda of=image.disk
```

Figure 74: dd Syntax



CERT Training and Education

Persistent Data Types: System Files

Microsoft Windows

- BIOS setup
  - Master boot record
  - Volume boot sector
- Event/system logs
  - MAC times
- System registry

- BIOS setup
  - Master boot record
  - Volume boot sector
- Event/system logs
  - /etc/syslog.conf
- Timestamps
  - Modification mtime
  - Access atime
  - Change of status ctime
  - Deleted time

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.7 Persistent Data Types

4.7.1 System Files

4.7.1.1 Windows

Basic Input/Output System (BIOS) Setup

- Stored in ROM and executed when the computer boots, these commands instruct the computer on how to read the physical disks.
- The partition table and boot information are on dedicated tracks. These tracks are not visible to the file utilities that normally access the partitions or file systems.
- An analysis of a complete hard drive image with a forensic tool or hex editor is necessary.

Boot Records

A single hard drive can be organized into multiple partitions. Because each partition could have a different OS, there are two types of boot records.

Master Boot Record (MBR)

The MBR is located on the first sector of the physical disk. It contains the information the computer needs to find the partition table and determine the OS. Accessing this boot code is one of the first things the computer does.

Volume Boot Sector

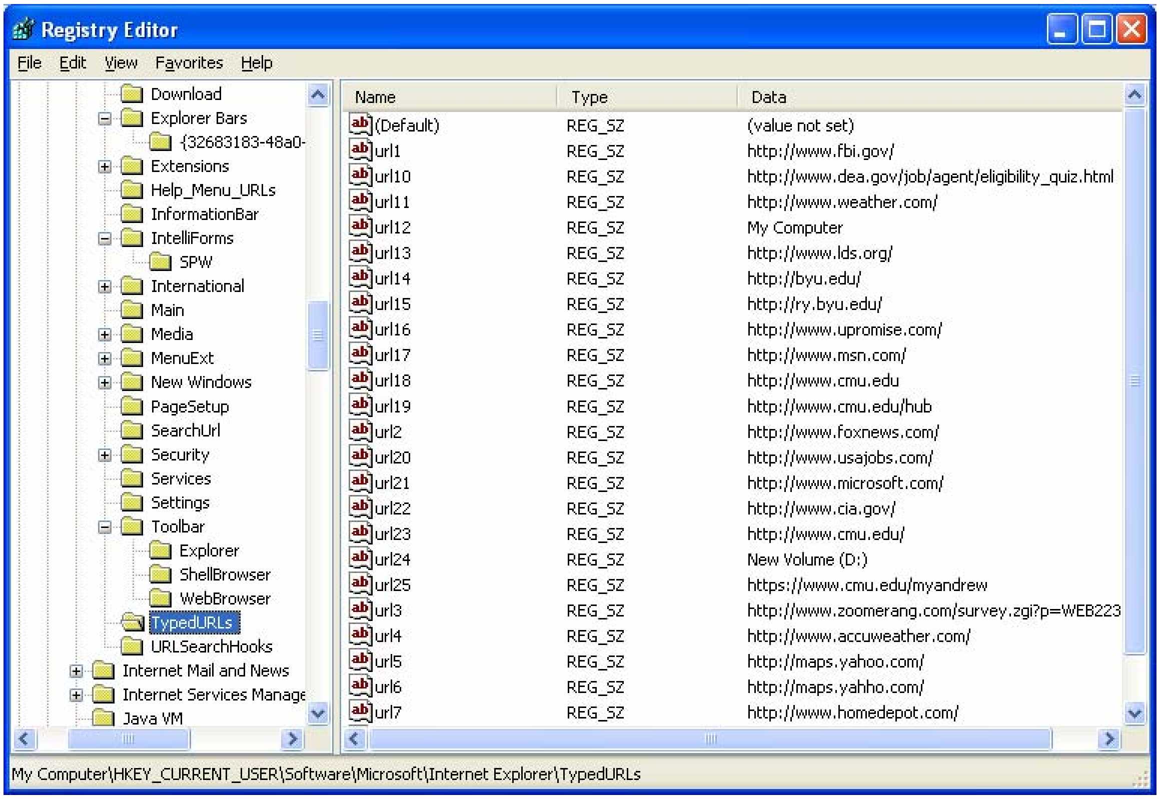
Each partition has a volume boot sector. This is used to specify the file system of that partition and provide any necessary boot code needed to mount the file system.

Event/System Logs

The key to getting useful logs is to collect them before a security incident occurs. In this case, a well developed policy pays off.

System Registry

Registry data is a source for users' settings and activities. For example, HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\TypedURLS provides a list of the typed URLs.



Name	Type	Data
(Default)	REG_SZ	(value not set)
url1	REG_SZ	http://www.fbi.gov/
url10	REG_SZ	http://www.dea.gov/job/agent/eligibility_quiz.html
url11	REG_SZ	http://www.weather.com/
url12	REG_SZ	My Computer
url13	REG_SZ	http://www.lds.org/
url14	REG_SZ	http://byu.edu/
url15	REG_SZ	http://ry.byu.edu/
url16	REG_SZ	http://www.upromise.com/
url17	REG_SZ	http://www.msn.com/
url18	REG_SZ	http://www.cmu.edu
url19	REG_SZ	http://www.cmu.edu/hub
url2	REG_SZ	http://www.foxnews.com/
url20	REG_SZ	http://www.usajobs.com/
url21	REG_SZ	http://www.microsoft.com/
url22	REG_SZ	http://www.cia.gov/
url23	REG_SZ	http://www.cmu.edu/
url24	REG_SZ	New Volume (D:)
url25	REG_SZ	https://www.cmu.edu/myandrew
url3	REG_SZ	http://www.zoomerang.com/survey.zgi?p=WEB223
url4	REG_SZ	http://www.accuweather.com/
url5	REG_SZ	http://maps.yahoo.com/
url6	REG_SZ	http://maps.yahho.com/
url7	REG_SZ	http://www.homedepot.com/

My Computer\HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\TypedURLs

Figure 75: The TypedURLs Folder

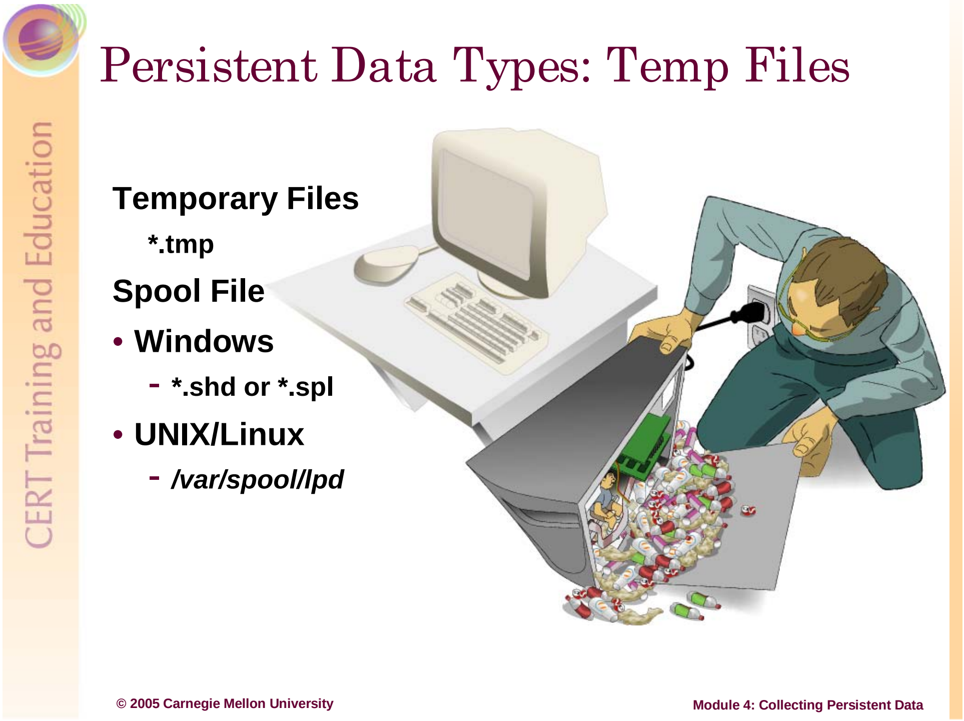
4.7.1.2 UNIX/Linux

BIOS Setup

Generally the boot sector is found in */dev/hda* and the file system is found in */dev/hda1*. A SCSI formatted drive might refer to this as */dev/sda* instead.

Event/System Logs

Almost any event on a Linux system can create a log entry. The degree of logging depends on the configuration of the system.



Persistent Data Types: Temp Files

Temporary Files

***.tmp**

Spool File

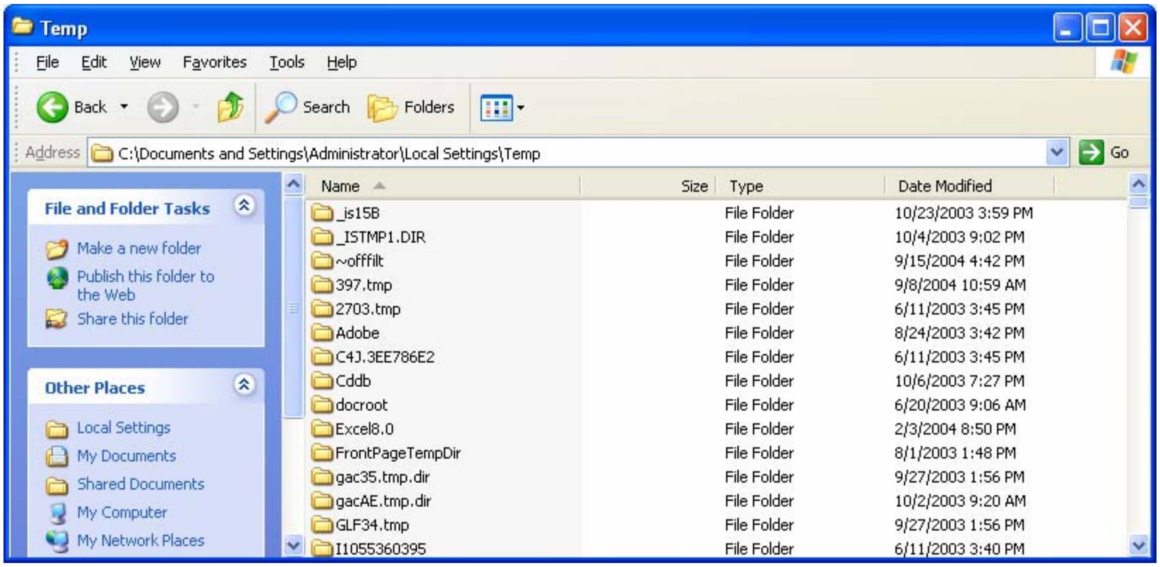
- **Windows**
  - ***.shd or *.spl**
- **UNIX/Linux**
  - ***/var/spool/lpd***

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.7.2 Temp Files

Temp (.tmp) files are created by many applications. Consider Microsoft Word: When you open a document, Word creates a temporary copy of the document to record all changes. When you are finished, Word saves all the recorded changes from the .tmp file into the original file. The .tmp file is then deleted and becomes a part of your hard drive's unallocated space [Kuepper 02].



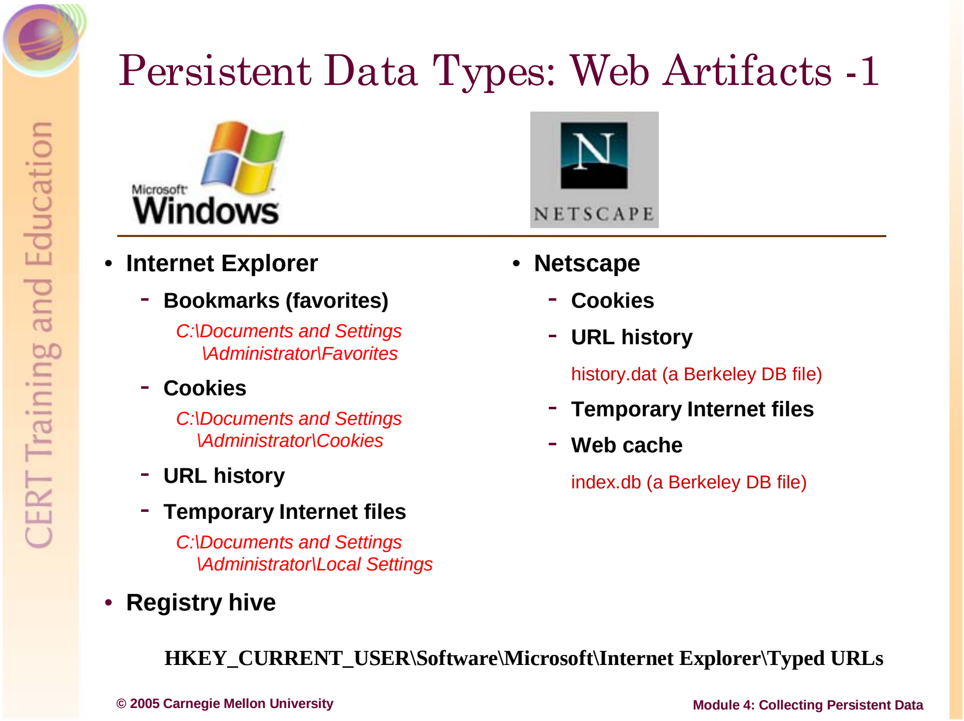
Address: C:\Documents and Settings\Administrator\Local Settings\Temp

Name	Size	Type	Date Modified
_is15B		File Folder	10/23/2003 3:59 PM
_ISTMP1.DIR		File Folder	10/4/2003 9:02 PM
~offfilt		File Folder	9/15/2004 4:42 PM
397.tmp		File Folder	9/8/2004 10:59 AM
2703.tmp		File Folder	6/11/2003 3:45 PM
Adobe		File Folder	8/24/2003 3:42 PM
C4J.3EE786E2		File Folder	6/11/2003 3:45 PM
Cddb		File Folder	10/6/2003 7:27 PM
docroot		File Folder	6/20/2003 9:06 AM
Excel8.0		File Folder	2/3/2004 8:50 PM
FrontPageTempDir		File Folder	8/1/2003 1:48 PM
gac35.tmp.dir		File Folder	9/27/2003 1:56 PM
gacAE.tmp.dir		File Folder	10/2/2003 9:20 AM
GLF34.tmp		File Folder	9/27/2003 1:56 PM
I1055360395		File Folder	6/11/2003 3:40 PM

Figure 76: Sample Temp Folder Contents

Another type of temp file is the spool file. As a job is sent to the printer, the spool file is created to store the information as it waits to be printed. After the job is printed, the disk space that was used by the spool file is reassigned as unallocated, but the printed file is intact until that disk space is overwritten.

Windows creates additional temp files during the printing process as described by Morris [Morris 03]. “These files, which have an .spl or .shd extension, contain a wealth of information about a print job, such as the name of the file printed, the owner of the file, the printer used, and the data to be printed (or a list of the temporary files containing such data).”



CERT Training and Education

Persistent Data Types: Web Artifacts -1

Microsoft Windows
NETSCAPE

- Internet Explorer
  - Bookmarks (favorites)
    C:\Documents and Settings \Administrator\Favorites
  - Cookies
    C:\Documents and Settings \Administrator\Cookies
  - URL history
  - Temporary Internet files
    C:\Documents and Settings \Administrator\Local Settings
- Registry hive

  HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Typed URLs

- Netscape
  - Cookies
  - URL history
    history.dat (a Berkeley DB file)
  - Temporary Internet files
  - Web cache
    index.db (a Berkeley DB file)

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.7.3 Web Artifacts

4.7.3.1 Windows vs. Linux

Web artifact data storage is not as uniform in Linux as it is for IE in Windows. Each Linux flavor, build, or browser might put the data in a slightly different place. Netscape stores the history in Berkeley DB file called history.dat. The cache is stored in another Berkeley DB file, index.db.

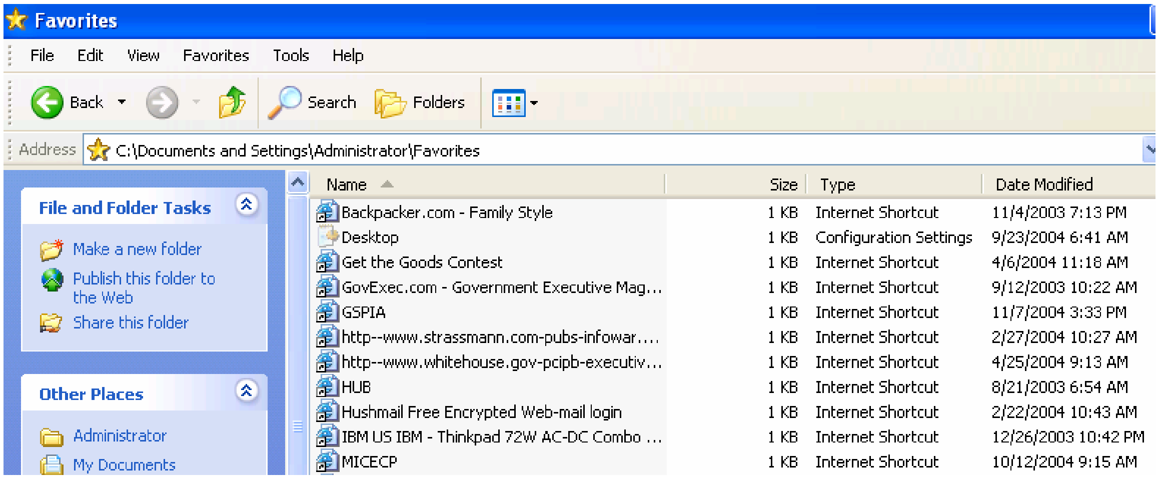
Before a security event occurs, then, be sure to document the locations of these files on all flavors of Linux.

In the following section, IE default locations are included with each artifact type.

4.7.3.2 IE Default Locations

Bookmarks

The IE default location is *C:\Documents and Settings\[user]\Favorites*.



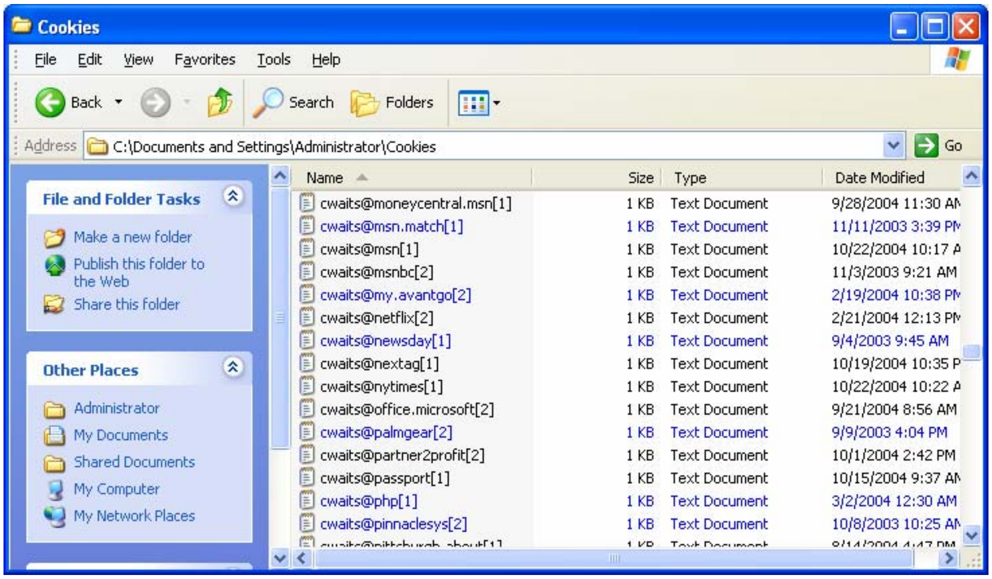
C:\Documents and Settings\Administrator\Favorites

Name	Size	Type	Date Modified
Backpacker.com - Family Style	1 KB	Internet Shortcut	11/4/2003 7:13 PM
Desktop	1 KB	Configuration Settings	9/23/2004 6:41 AM
Get the Goods Contest	1 KB	Internet Shortcut	4/6/2004 11:18 AM
GovExec.com - Government Executive Mag...	1 KB	Internet Shortcut	9/12/2003 10:22 AM
GSPIA	1 KB	Internet Shortcut	11/7/2004 3:33 PM
http--www.strassmann.com-pubs-infowar....	1 KB	Internet Shortcut	2/27/2004 10:27 AM
http--www.whitehouse.gov-pcipb-executiv...	1 KB	Internet Shortcut	4/25/2004 9:13 AM
HUB	1 KB	Internet Shortcut	8/21/2003 6:54 AM
Hushmail Free Encrypted Web-mail login	1 KB	Internet Shortcut	2/22/2004 10:43 AM
IBM US IBM - Thinkpad 72W AC-DC Combo ...	1 KB	Internet Shortcut	12/26/2003 10:42 PM
MICECP	1 KB	Internet Shortcut	10/12/2004 9:15 AM

Figure 77: The IE Default Location for Bookmarks

Cookies

The IE default location is *Documents and Settings\[user]\Cookies*.



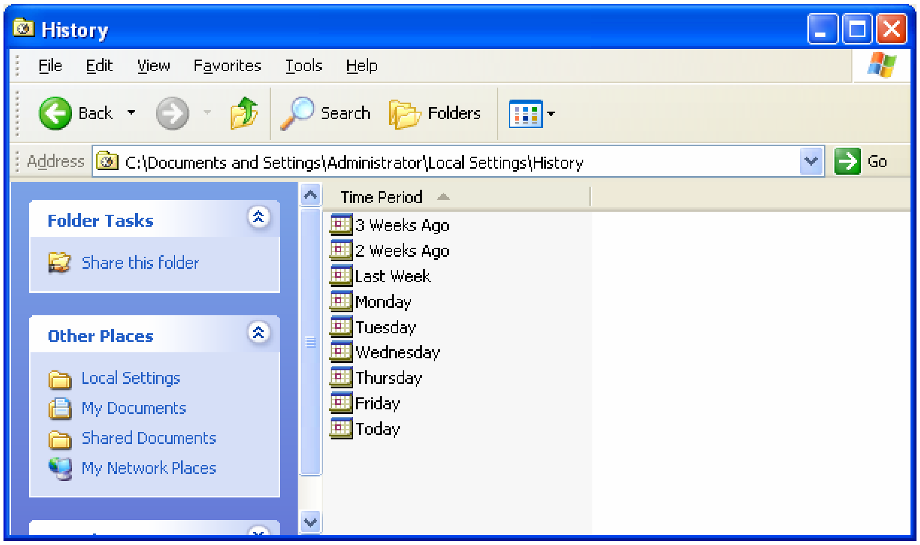
C:\Documents and Settings\Administrator\Cookies

Name	Size	Type	Date Modified
cwaits@moneycentral.msn[1]	1 KB	Text Document	9/28/2004 11:30 AM
cwaits@msn.match[1]	1 KB	Text Document	11/11/2003 3:39 PM
cwaits@msn[1]	1 KB	Text Document	10/22/2004 10:17 A
cwaits@msnbc[2]	1 KB	Text Document	11/3/2003 9:21 AM
cwaits@my.avantgo[2]	1 KB	Text Document	2/19/2004 10:38 PM
cwaits@netflix[2]	1 KB	Text Document	2/21/2004 12:13 PM
cwaits@newsday[1]	1 KB	Text Document	9/4/2003 9:45 AM
cwaits@nextag[1]	1 KB	Text Document	10/19/2004 10:35 P
cwaits@nytimes[1]	1 KB	Text Document	10/22/2004 10:22 A
cwaits@office.microsoft[2]	1 KB	Text Document	9/21/2004 8:56 AM
cwaits@palmgear[2]	1 KB	Text Document	9/9/2003 4:04 PM
cwaits@partner2profit[2]	1 KB	Text Document	10/1/2004 2:42 PM
cwaits@passport[1]	1 KB	Text Document	10/15/2004 9:37 AM
cwaits@php[1]	1 KB	Text Document	3/2/2004 12:30 AM
cwaits@pinnaclesys[2]	1 KB	Text Document	10/8/2003 10:25 AM
cwaits@pittsburgh_about[1]	1 KB	Text Document	9/14/2004 4:47 PM

Figure 78: The IE Default Location for Cookies

URL History

The IE default location is *Documents and Settings\[user]\Local Settings\History.*



Address: C:\Documents and Settings\Administrator\Local Settings\History

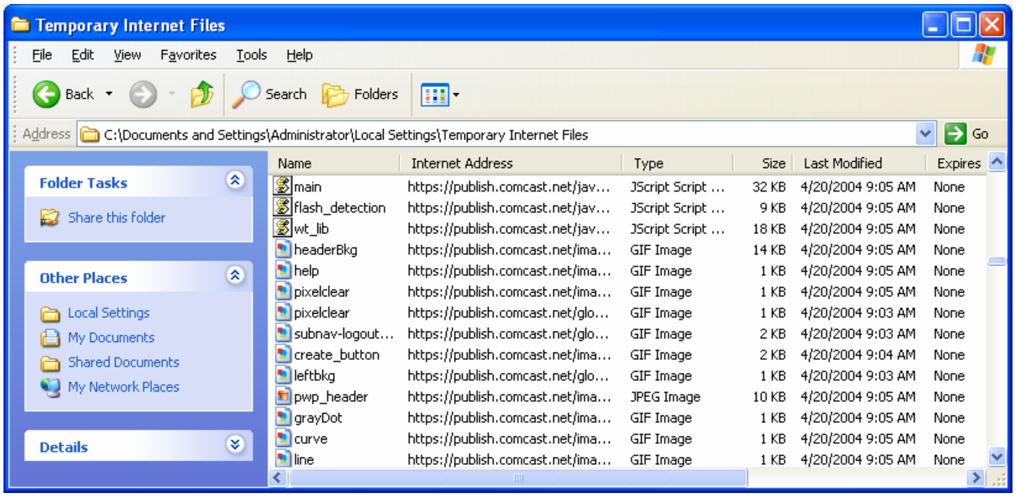
Time Period
3 Weeks Ago
2 Weeks Ago
Last Week
Monday
Tuesday
Wednesday
Thursday
Friday
Today

Figure 79: The IE Default Location for URL History

Temporary Internet Files/Web Cache

The contents of each Web page you visit are downloaded to files. Even after these files are deleted, information is left in the unallocated space.

The IE default location is *\Documents and Settings\[user]\Local Settings\Temporary Internet Files.*



Address: C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files

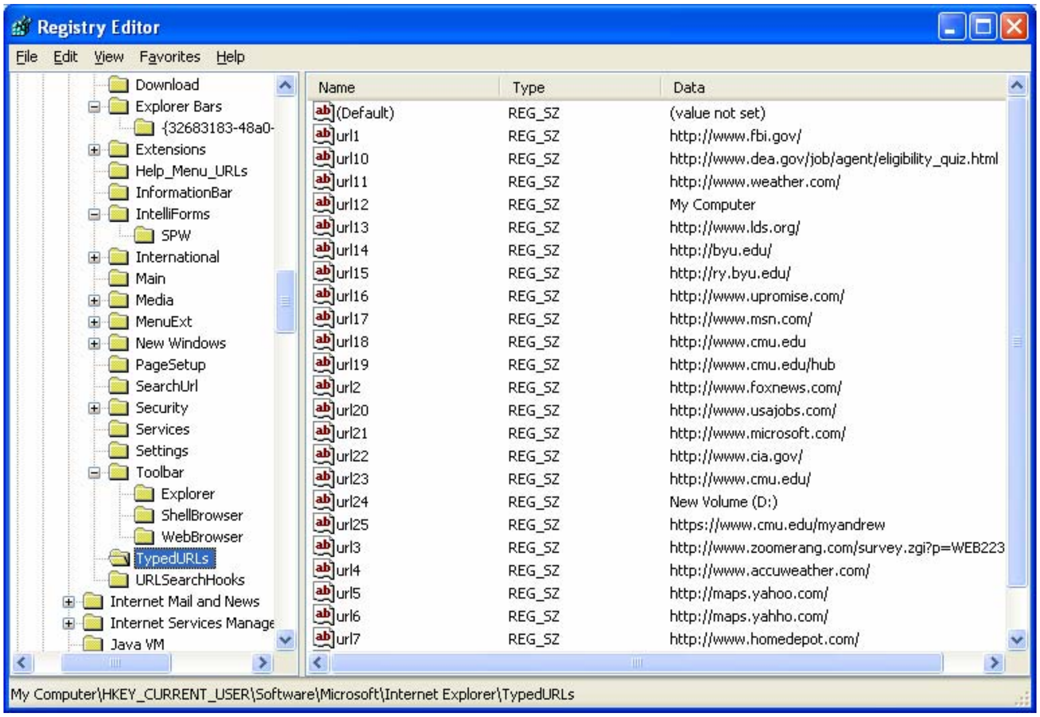
Name	Internet Address	Type	Size	Last Modified	Expires
main	https://publish.comcast.net/jav...	JScript Script ...	32 KB	4/20/2004 9:05 AM	None
flash_detection	https://publish.comcast.net/jav...	JScript Script ...	9 KB	4/20/2004 9:05 AM	None
wt_lib	https://publish.comcast.net/jav...	JScript Script ...	18 KB	4/20/2004 9:05 AM	None
headerBkg	https://publish.comcast.net/ima...	GIF Image	14 KB	4/20/2004 9:05 AM	None
help	https://publish.comcast.net/ima...	GIF Image	1 KB	4/20/2004 9:05 AM	None
pixelclear	https://publish.comcast.net/ima...	GIF Image	1 KB	4/20/2004 9:05 AM	None
pixelclear	https://publish.comcast.net/glo...	GIF Image	1 KB	4/20/2004 9:03 AM	None
subnav-logout...	https://publish.comcast.net/glo...	GIF Image	2 KB	4/20/2004 9:03 AM	None
create_button	https://publish.comcast.net/ima...	GIF Image	2 KB	4/20/2004 9:04 AM	None
leftbkg	https://publish.comcast.net/glo...	GIF Image	1 KB	4/20/2004 9:03 AM	None
pwp_header	https://publish.comcast.net/ima...	JPEG Image	10 KB	4/20/2004 9:05 AM	None
grayDot	https://publish.comcast.net/ima...	GIF Image	1 KB	4/20/2004 9:05 AM	None
curve	https://publish.comcast.net/ima...	GIF Image	1 KB	4/20/2004 9:05 AM	None
line	https://publish.comcast.net/ima...	GIF Image	1 KB	4/20/2004 9:05 AM	None

Figure 80: The IE Default Location for Web Cache

Registry Hive

The registry HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Typed URLs contains a record of the URLs typed into IE. If a user goes to the history file and deletes entries, those URLs may still be in the registry hive. However, if a user clicks the “Clear

History" button in the Tools > Internet Options > "General" tab of IE, the entries in the hive will be cleared out.

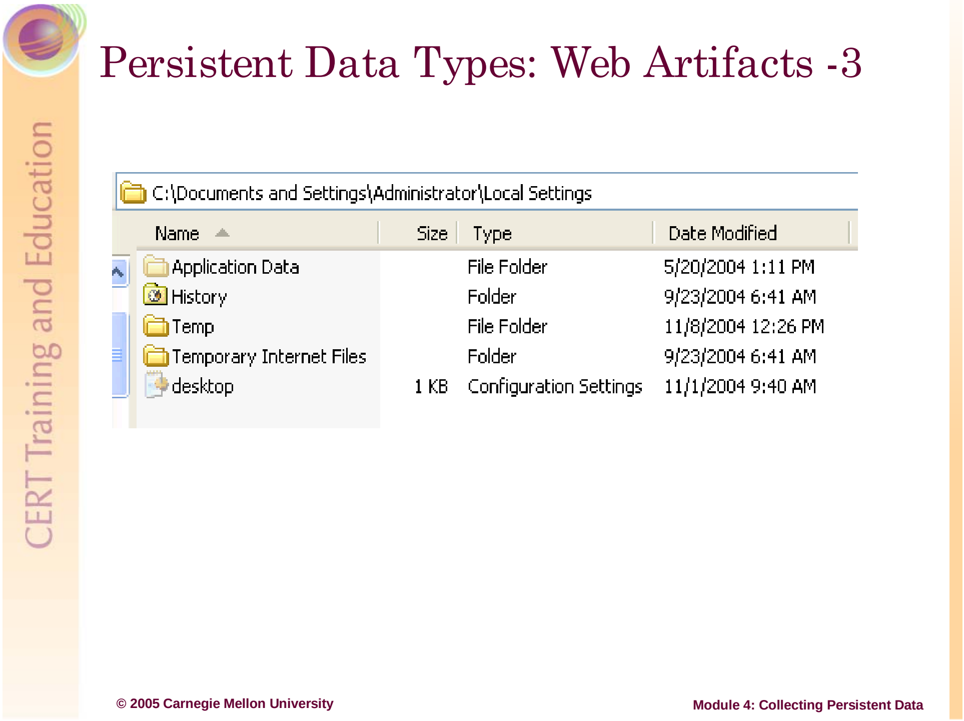


Name	Type	Data
(Default)	REG_SZ	(value not set)
url1	REG_SZ	http://www.fbi.gov/
url10	REG_SZ	http://www.dea.gov/job/agent/eligibility_quiz.html
url11	REG_SZ	http://www.weather.com/
url12	REG_SZ	My Computer
url13	REG_SZ	http://www.lds.org/
url14	REG_SZ	http://byu.edu/
url15	REG_SZ	http://ry.byu.edu/
url16	REG_SZ	http://www.upromise.com/
url17	REG_SZ	http://www.msn.com/
url18	REG_SZ	http://www.cmu.edu
url19	REG_SZ	http://www.cmu.edu/hub
url2	REG_SZ	http://www.foxnews.com/
url20	REG_SZ	http://www.usajobs.com/
url21	REG_SZ	http://www.microsoft.com/
url22	REG_SZ	http://www.cia.gov/
url23	REG_SZ	http://www.cmu.edu/
url24	REG_SZ	New Volume (D:)
url25	REG_SZ	https://www.cmu.edu/myandrew
url3	REG_SZ	http://www.zoomerang.com/survey.zgi?p=WEB223
url4	REG_SZ	http://www.accuweather.com/
url5	REG_SZ	http://maps.yahoo.com/
url6	REG_SZ	http://maps.yahho.com/
url7	REG_SZ	http://www.homedepot.com/

My Computer\HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\TypedURLs

4.7.3.3 Alternative Browsers

Remember that though we live in a Windows universe, there are other browsers. The cookies, history, cache, and temp files for the Web activity performed using these other browsers are not stored in the Windows locations. Where you find this data depends on how the browser was installed. For instance, the Mozilla browser installed on my computer stores the cookies, history, and cache in the *C:\Documents and Settings\Administrator\Application Data\Mozilla\Profiles\default\wq809fmw.slt* folder.



CERT Training and Education

Persistent Data Types: Web Artifacts -3

C:\Documents and Settings\Administrator\Local Settings

Name	Size	Type	Date Modified
Application Data		File Folder	5/20/2004 1:11 PM
History		Folder	9/23/2004 6:41 AM
Temp		File Folder	11/8/2004 12:26 PM
Temporary Internet Files		Folder	9/23/2004 6:41 AM
desktop	1 KB	Configuration Settings	11/1/2004 9:40 AM

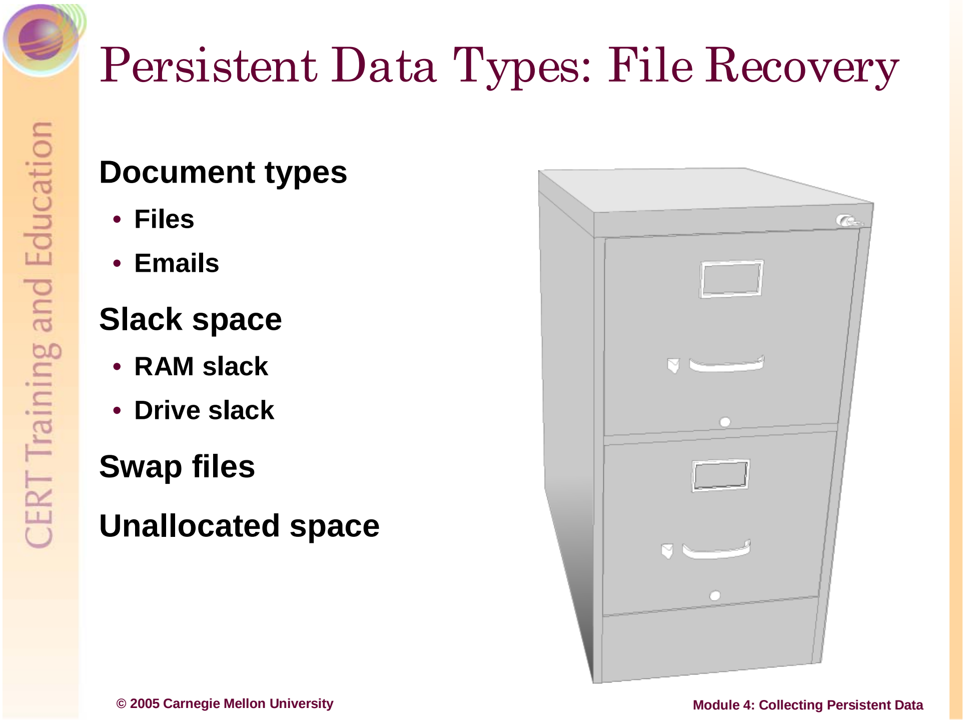
© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.7.3.4 Cookies

Often, a Web site gathers information about a visitor’s habits by placing a cookie on the user’s computer. These can be used for anything from keeping track of user’s preferred book genre in order to make better recommendations to remembering the user’s zip code so they can provide local weather reports.

Cookies can also be used for nefarious purposes. Malicious Web sites can upload malware masquerading as a cookie to a user’s computer. For more information on cookies and their uses, visit http://www.cookiecentral.com/.



CERT Training and Education

Persistent Data Types: File Recovery

Document types

- Files
- Emails

Slack space

- RAM slack
- Drive slack

Swap files

Unallocated space

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.7.4 File Recovery

While there are some differences, Windows and Linux file recovery is similar.

4.7.4.1 Deleted Data

There are several different ways to delete data in Windows.

Files can be deleted using Windows Explorer, which is the most common method. The deleted file is put into the Recycle Bin but can be easily restored because, per Nelson [Nelson 04], Windows first “changes the name of the file and moves it to a subdirectory with a unique identity in the Recycle Bin.” Info2, the control file for the Recycle Bin, contains information about the original path and file name.

Files can be deleted using the MS-DOS functions.

Files can be “wiped” using a variety of special software tools. Remember from Module 2 that wiping is different than deleting because clusters where the data was stored are written over to prevent their recovery from slack or unallocated space.

Deleting a file, however, does not remove its short cut from the Recent Files list. These files, which are not available through normal means, are available through a forensic examination of the slack or unallocated space.

Not all flavors of Linux make use of a Recycle bin the way Windows does. When deleted on a UNIX system, Casey [Casey 00] says “the file’s directory entry is hidden from view and the system notes that the associated inode is available for reuse. The file's directory entry, inode, and data remain on the disk until they are overwritten."

4.7.4.2 Slack Space

Logical file size: Actual size of the file

Physical file size: Amount of space a file takes up on the disk, given in whole number of clusters

Sector slack: Left over capacity in a sector, filled with data held in the RAM

Drive slack: Excess capacity in a cluster, empty or filled with whatever was previously stored in that series of consecutive sectors

4.7.4.3 Swap Files

Windows swap files can be temporary or permanent, depending on which Microsoft OS you have.

In Windows 95, 98, and Me, the temporary swap file expands and contracts depending on how much space it needs. To run multiple programs simultaneously with limited RAM, Windows uses this swap file to expand the computer's RAM. Windows swaps an application's data to and from the swap file as needed, based on what is actively running. Because this file contains application data, there could be almost any type of data in it: passwords, complete or partial documents, email, credit card numbers, images, and so on.

In Windows NT, 2000, and XP, the permanent swap file does not expand and contract. By default, Windows 2000 sets the swap size to the amount of RAM * 1.5.

> *As a side note, there is a local security policy setting under Start > Programs > Administrative Tools > Local Security Policies > Local Policies > Security Options that allows you to set Windows to "Clear virtual memory page when Windows shuts down" [Kuepper 02].*

4.7.4.4 Unallocated Space

Unallocated space are portions of the drive that are not actively assigned data.

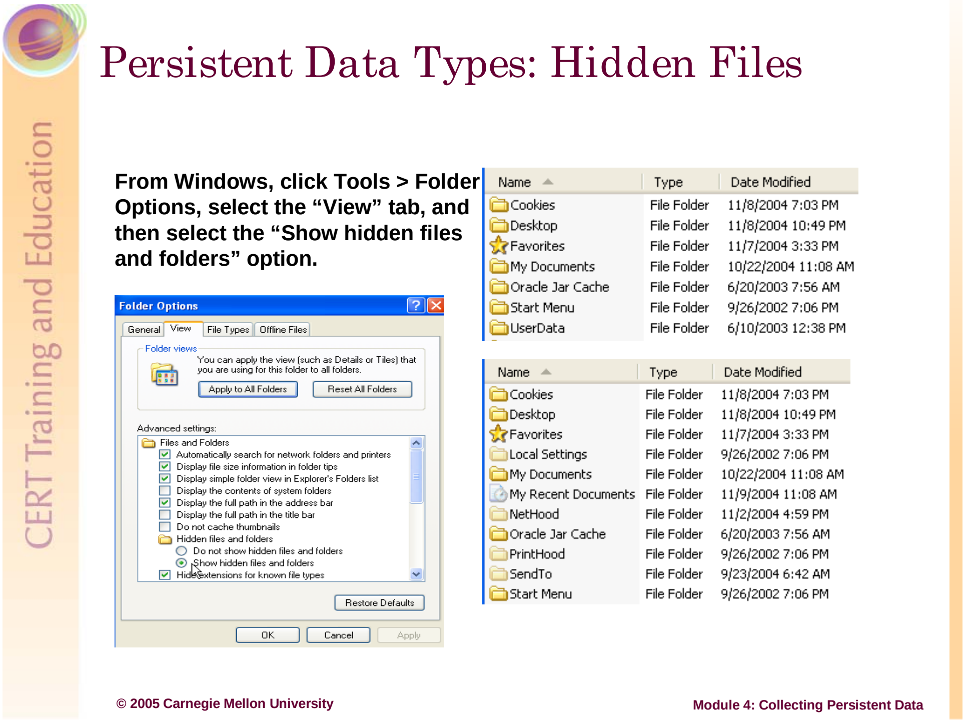
4.7.4.5 Partial Files

At times, the file system begins writing a file to a disk or partition and does not realize that the target space can not contain the entire file until it has filled all the available space. This generally results in an error and a request that the user clear some space. Partial files do not show up in the directory but remain in unallocated space until they are overwritten.

4.7.4.6 Windows Artifacts

Windows artifacts, which are created by the Windows OS, are important for forensic examiners as they attempt to recreate a user’s activities—partly because the artifacts are often overlooked by users or intruders as they attempt to cover their tracks. They include files used as shortcuts to applications, files, or devices, or certain files created by the OS. Examples are the .lnk files created in the Windows Desktop, Recent, Send To, and Start menu folders.

Examine these artifacts to uncover files, folders, applications, or devices that are no longer on the system. For example, a forensic examiner may discover links to files that have been deleted and are no longer recoverable or that were stored on a network drive. Links to other storage devices (i.e., Zip or Jazz drive) may indicate that other media need to be located and analyzed [Morris 03].



CERT Training and Education

Persistent Data Types: Hidden Files

From Windows, click Tools > Folder Options, select the "View" tab, and then select the "Show hidden files and folders" option.

Folder Options

General | View | File Types | Offline Files

Folder views

You can apply the view (such as Details or Tiles) that you are using for this folder to all folders.

Apply to All Folders | Reset All Folders

Advanced settings:

- Files and Folders
  - ☑ Automatically search for network folders and printers
  - ☑ Display file size information in folder tips
  - ☑ Display simple folder view in Explorer's Folders list
  - □ Display the contents of system folders
  - ☑ Display the full path in the address bar
  - □ Display the full path in the title bar
  - □ Do not cache thumbnails
  - Hidden files and folders
    - ○ Do not show hidden files and folders
    - ● Show hidden files and folders
  - ☑ Hide extensions for known file types

Restore Defaults

OK | Cancel | Apply

Name	Type	Date Modified
Cookies	File Folder	11/8/2004 7:03 PM
Desktop	File Folder	11/8/2004 10:49 PM
Favorites	File Folder	11/7/2004 3:33 PM
My Documents	File Folder	10/22/2004 11:08 AM
Oracle Jar Cache	File Folder	6/20/2003 7:56 AM
Start Menu	File Folder	9/26/2002 7:06 PM
UserData	File Folder	6/10/2003 12:38 PM

Name	Type	Date Modified
Cookies	File Folder	11/8/2004 7:03 PM
Desktop	File Folder	11/8/2004 10:49 PM
Favorites	File Folder	11/7/2004 3:33 PM
Local Settings	File Folder	9/26/2002 7:06 PM
My Documents	File Folder	10/22/2004 11:08 AM
My Recent Documents	File Folder	11/9/2004 11:08 AM
NetHood	File Folder	11/2/2004 4:59 PM
Oracle Jar Cache	File Folder	6/20/2003 7:56 AM
PrintHood	File Folder	9/26/2002 7:06 PM
SendTo	File Folder	9/23/2004 6:42 AM
Start Menu	File Folder	9/26/2002 7:06 PM

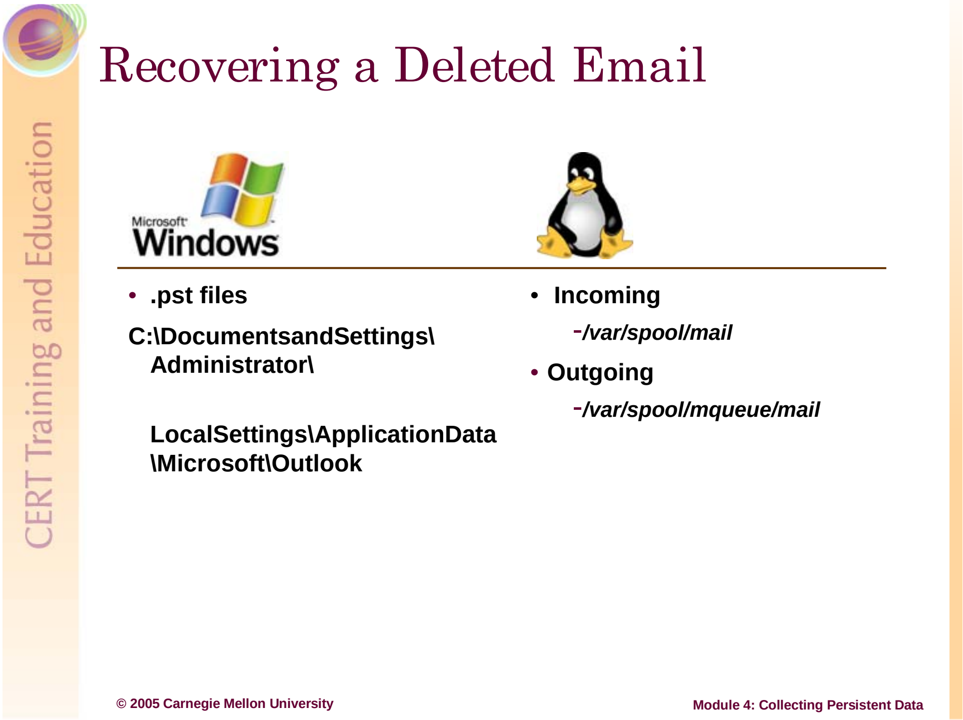
© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.7.5 Hidden Files

Files can be assigned an attribute so that they will not be displayed with normal file system viewing methods (i.e., Windows Explorer and the DOS *dir* command). Files may be marked as hidden to protect files from being corrupted by casual users or to hide illicit data or activities.

To view hidden files with Windows Explorer, go to Tools > Folder Options, click the "View" tab, and select the Show Hidden files option.


CERT Training and Education
Microsoft Windows

Recovering a Deleted Email

- .pst files

C:\DocumentsandSettings\
Administrator\

LocalSettings\ApplicationData
\Microsoft\Outlook

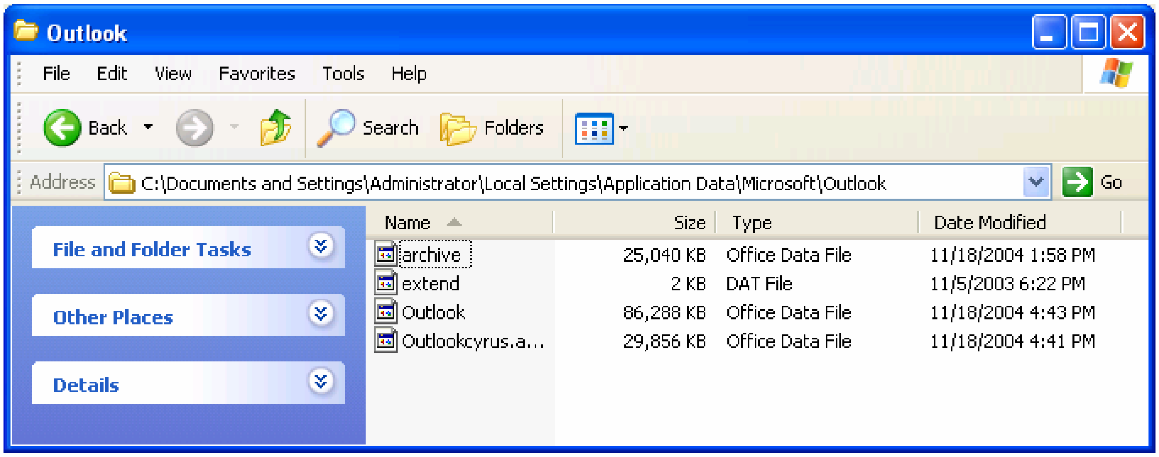
- Incoming
  - */var/spool/mail*
- Outgoing
  - */var/spool/mqueue/mail*

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.8 Recovering a Deleted Email

Windows Outlook emails are .pst files. Their default location is *\Documents and Settings\[user]\Local Settings\Application Data\Microsoft\Outlook.*

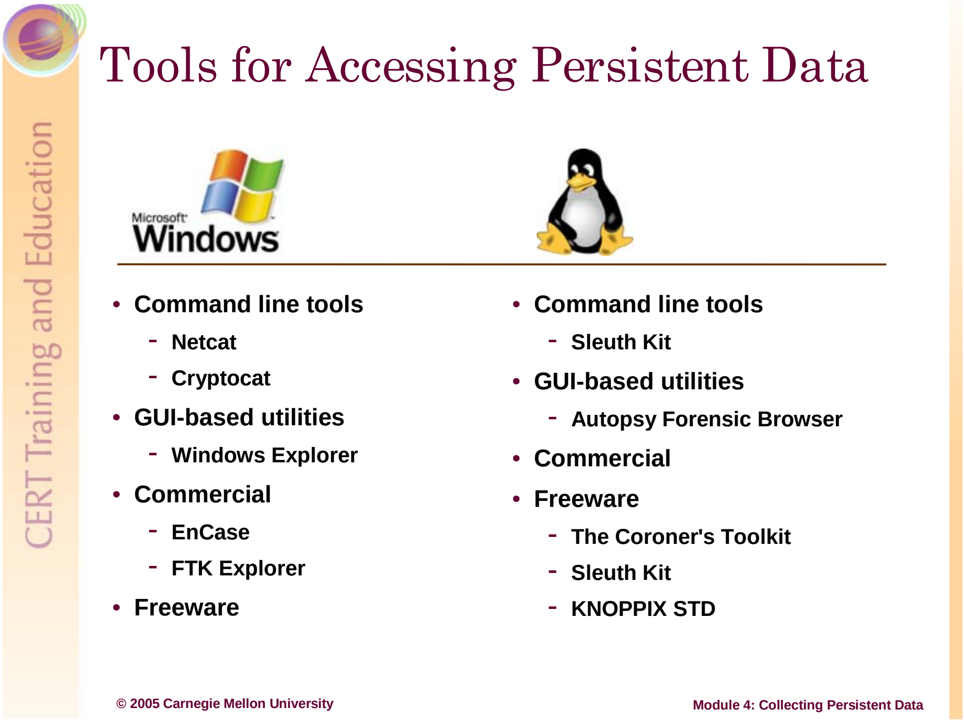


Address C:\Documents and Settings\Administrator\Local Settings\Application Data\Microsoft\Outlook

Name	Size	Type	Date Modified
archive	25,040 KB	Office Data File	11/18/2004 1:58 PM
extend	2 KB	DAT File	11/5/2003 6:22 PM
Outlook	86,288 KB	Office Data File	11/18/2004 4:43 PM
Outlookcyrus.a...	29,856 KB	Office Data File	11/18/2004 4:41 PM

Figure 81: The Default Location for Outlook Email Files

Because the .pst files are in a proprietary Windows format, they are more easily viewed with Outlook or Outlook Express. To view them, copy the .pst files from the suspicious computer to another computer with Outlook.



CERT Training and Education

Tools for Accessing Persistent Data

- Command line tools
  - Netcat
  - Cryptocat
- GUI-based utilities
  - Windows Explorer
- Commercial
  - EnCase
  - FTK Explorer
- Freeware

- Command line tools
  - Sleuth Kit
- GUI-based utilities
  - Autopsy Forensic Browser
- Commercial
- Freeware
  - The Coroner's Toolkit
  - Sleuth Kit
  - KNOPPIX STD

© 2005 Carnegie Mellon University

Module 4: Collecting Persistent Data

4.9 Tools for Accessing Persistent Data

There is a lot of overlap in the forensic tool market. The most important quality to look for is usability. If the tool works for you and it meets admissibility standards, select it.

Command-line tools are generally small and portable. They fit easily onto a floppy disk or CD-ROM and if packaged correctly can be used safely on a live suspicious computer.

Commercial tools are more polished and often more steamlined. They perform the same tasks as the freeware tools, but what might take 13 steps with a free product will only take two steps with a commercial product.

What follows is a small sample of the forensic tools available.

4.9.1 Windows

4.9.1.1 Command-Line Tools

- Netcat
- dd

  Download this tool from http://uranus.it.swin.edu.au/~jn/linux/rawwrite/dd.htm.

4.9.1.2 GUI-Based Utilities

- Windows Explorer

Explorer is native to the Windows OS. It allows you to navigate through the file systems of a mounted image drive.

4.9.1.3 Commercial

- EnCase

  EnCase is a full service tool that handles acquisition and analysis of volatile and non-volatile data.

  For more details, visit http://www.guidancesoftware.com/products/index.shtm.

- FTK (Forensic Tool Kit)

  FTK is an all-in-one collection and examination tool.

  For more details, visit http://www.accessdata.com/Product04_Overview.htm?ProductNum=04.

4.9.2 UNIX/Linux

4.9.2.1 Command-Line Tools

- Netcat

  Netcat is a networking utility that reads and writes data across network connections using the TCP/IP protocol.

  For more details, visit http://netcat.sourceforge.net.

- dd

  This tools creates drive or partition images. A version of this tool has also been developed for Windows. Most versions of Linux have *dd* as a native tool.

4.9.2.2 GUI-Based Utilities

- Autopsy Forensic Browser

  The Autopsy Forensic Browser is a graphical interface to the command-line digital forensic analysis tool, The Sleuth Kit. Together, The Sleuth Kit and Autopsy provide many of the same features as commercial digital forensics tools for the analysis of Windows and UNIX file systems (NTFS, FAT, FFS, ext2, and ext3).

  For more details, visit http://www.sleuthkit.org/autopsy

4.9.2.3 Freeware

- The Sleuth Kit

  The Sleuth Kit (previously known as TASK) is a collection of UNIX-based command-line file system and media management forensic analysis tools. The file system tools allow you to examine a suspicious computer in a non-intrusive fashion. Because the tools do not rely on the OS to process the file systems, deleted and hidden content is shown.

For more details, visit http://www.sleuthkit.org/sleuthkit/desc.php.

- The Coroner’s Toolkit

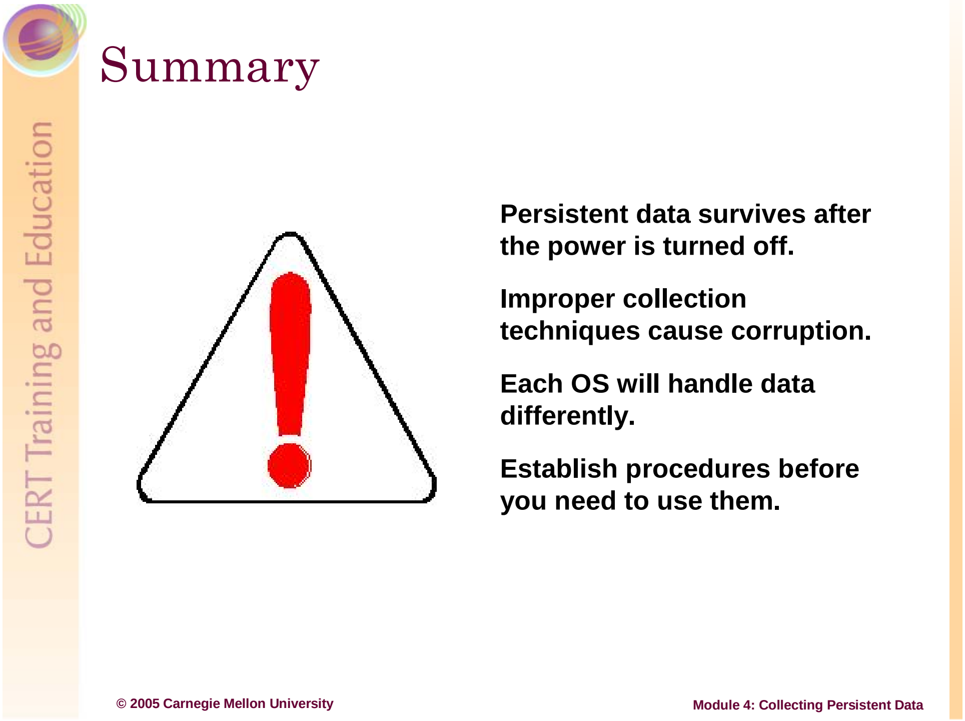
  The Coroner’ Toolkit (TCT) is a collection of tools that gather and analyze data on a UNIX. `unrm` and `lazarus` are part of the toolkit and can be used to restore deleted files and data that cannot be easily accessed.

  For more details, visit http://www.fish.com/tct.

- Knoppix STD

  Knoppix-STD is a customized distribution of the Knoppix Live Linux CD. Boot to the CD and you have Knoppix-STD. STD focuses on information security and network management tools.

  For more details, visit http://www.knoppix-std.org.



CERT Training and Education

Summary

Persistent data survives after the power is turned off.

Improper collection techniques cause corruption.

Each OS will handle data differently.

Establish procedures before you need to use them.

© 2005 Carnegie Mellon University

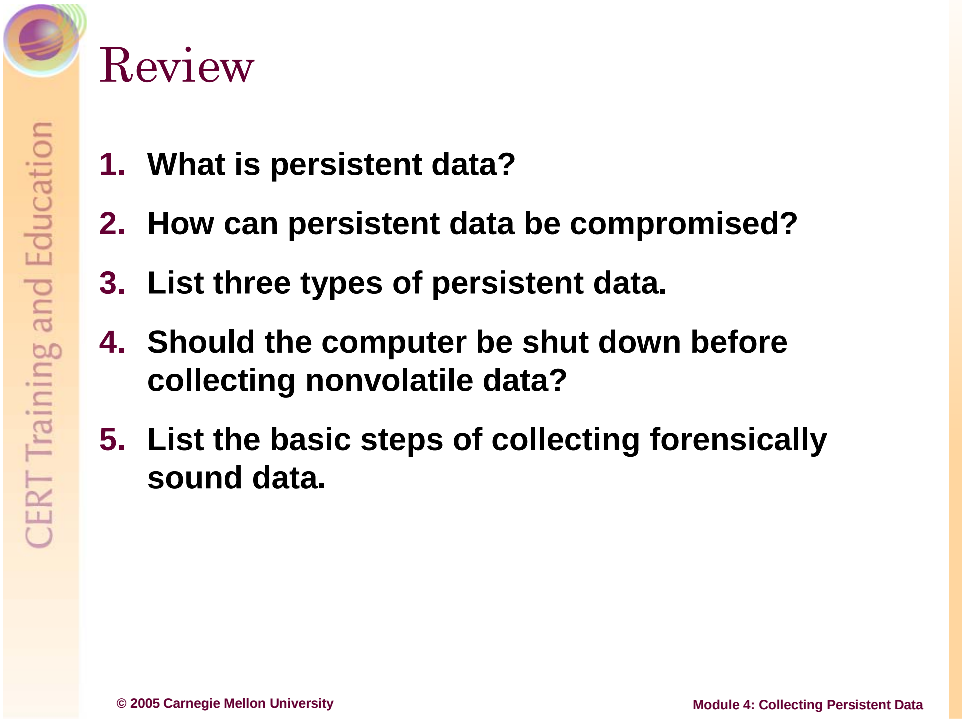
Module 4: Collecting Persistent Data

4.10 Summary

It is important to remember at least four points from this module:

1. Persistent data survives after the power is turned off.
2. Improper collection techniques cause corruption. “Primum non nocere” or “First do no harm” could be the dictum when collecting persistent data.
3. Each OS will handle data differently, so be familiar with the computers on your network.
4. Establish procedures before you need them.

   Performing an ad hoc investigation leads to unnecessary aggravation. There will be enough aggravation even with proper preparation.


CERT Training and Education

Review

1. **What is persistent data?**
2. **How can persistent data be compromised?**
3. **List three types of persistent data.**
4. **Should the computer be shut down before collecting nonvolatile data?**
5. **List the basic steps of collecting forensically sound data.**

Module 4: Collecting Persistent Data

4.11 Review

1. Persistent (or nonvolatile) data retains its state after the power has been turned off.
2. Compromised persistent data can result from just about any action taken on the host computer. Browsing the Web, opening files, and even creating new files can compromise persistent data.
3. Types of persistent data include deleted files, cookies, Web history, MAC times, registry hives, emails, recycle bin, and recent document shortcuts.
4. The computer should not necessarily be shut down before collecting nonvolatile data. It is always easier to collect persistent data from a dead hard drive mounted on a forensic workstation, but circumstances dictate the appropriate course of action. Sometimes a computer must be left on. In these cases there is nothing to do but collect from the live computer.
5. Steps for collecting forensically sound data include (1) have a pre-established procedure in place; (2) document steps taken; (3) image the suspicious computer’s hard drive; (4) establish the integrity of the image using a MD5 hash; (5) perform any analysis of working copies.

References

URLs are valid as of the publication date of this document.

[ACPO 03] Association of Chief Police Officers (ACPO). *Good Practice Guide for Computer based Electronic Evidence*. http://www.4law.co.il/Lea92.htm (2003).

[Barrett 04] Barrett, Daniel J. *Linux Pocket Guide*. Sebastopol, CA: O’Reilly, 2004.

[Brezinski 02] Brezinski, D. *Guidelines for Evidence Collection and Archiving* (Network Working Group RFC 3227). http://www.ietf.org/rfc/rfc3227.txt (2002).

[CAE 04] Computer-Aided Engineering Center, University of Wisconsin-Madison, College of Engineering. *Linux File Permissions Overview*. http://www.cae.wisc.edu/fsg/linux/linux-filepermissions.shtml (2004).

[Carvey 04] Carvey, Harlan. *Windows Forensics and Incident Recovery*. Boston, MA: Addison-Wesley, 2004 (ISBN 0-321-20098-5).

[Casey 00] Casey, Eoghan. *Digital Evidence and Computer Crime, Second Edition*. San Diego, CA: Academic Press, 2000 (ISBN 0-121-62885-X).

[CDT 04] Center for Democracy and Technology. *Impact of the McCain-Kerrey Bill on Constitutional Privacy Rights*. http://www.cdt.org/crypto/legis_105/mccain_kerrey/const_impact.html (2004).

[GNU 04] The GNU Netcat Project. *What is Netcat?* http://netcat.sourceforge.net (2004).

[Grundy 04] Grundy, Barry J. The Law Enforcement and Forensic Examiner Introduction to Linux: A Beginner’s Guide, Ver 2.0.5. http://www.linux-forensics.com/linuxintro-LEFE-2.0.5.pdf (2004).

[Haas 04] Haas, Juergen. *Linux / Unix Command: checkconfig*. http://linux.about.com/library/cmd/blcmdl8_chkconfig.htm (2004).

[Honeynet 04] The Honeynet Project. *Know Your Enemy: Learning About Security Threats, Second Edition*. Boston, MA: Pearson Education, Inc., 2004 (ISBN 0-321-16646-9).

[ICH 03] ICH Architecture Resource Center. *Interoperability Clearinghouse Glossary of Terms*. www.ichnet.org/glossary.htm (2003).

[Kerr 01] Kerr, Orin S. "Computer Records and the Federal Rules of Evidence." *USA Bulletin 49*, 2 (March 2001): 25-32. http://www.usdoj.gov/usao/eousa/foia_reading_room/usab4902.pdf

[Kuepper 02] Kuepper, Brian. *What You Don't See on Your Hard Drive*. http://www.sans.org/rr/whitepapers/incident/653.php (2002).

[Mandia 01] Mandia, Kevin & Prosise, Chris. *Incident Response: Investigating Computer Crime*. Berkeley, CA: Osborne/McGraw-Hill, 2001.

[Microsoft 03a] Microsoft. *FAT File System Technology and Patent License*. http://www.microsoft.com/mscorp/ip/tech/fat.asp (2003).

[Microsoft 03b] Microsoft. *FAT File System: The Story Behind the Innovation.* http://www.microsoft.com/mscorp/ip/tech/fathist.asp (2003).

[Microsoft 04] Microsoft. *Microsoft® Windows® XP Professional Resource Documentation Kit.* http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us (2004).

[MIT 03] Massachusetts Institute of Technology. *MIT researchers uncover mountains of private data on discarded computers*. http://web.mit.edu/newsoffice/2003/diskdrives.html (2003).

[Morris 03] Morris, Jamie. *Forensics on the Windows Platform, Part Two*. http://www.securityfocus.com/infocus/1665 (2003).

[Nelson 04] Nelson, Bill. *Guide to Computer Forensics and Investigations*. Boston, MA: Thomson Course Technology, 2004 (ISBN 0-619-13120-9).

[Nemeth 01] Nemeth, Evi. *UNIX System Administration Handbook, 3rd Edition*. Upper Saddle River, NJ: Prentice Hall PTR, 2001 (ISBN 0-130-20601-6).

[NIJ 04] National Institute of Justice. *Forensic Examination of Digital Evidence: A Guide for Law Enforcement.* http://www.ncjrs.org/pdffiles1/nij/199408.pdf (2004).

[NIST 01] Wack, John P. *Establishing a Computer Security Incident Response Capability*. http://csrc.nist.gov/publications/nistpubs/800-3/800-3.pdf (2001).

[NIST 03] National Institute of Standards and Technology. *CFTT Metholology Overview*. http://www.cftt.nist.gov/Methodology_Overview.htm (2003).

[NTI 03] New Technologies, Inc. *Computer Forensics Definitions.* http://www.forensics-intl.com/define.html (2003).

[Pierce 03] Pierce, Matt. *Detailed Forensic Procedure for Laptop Computers*. http://www.giac.org/practical/GSEC/Matt_Pierce_GSEC.pdf (2003).

[Russinovich 97] Russinovich, Mark. *Inside the Registry.* http://www.microsoft.com/technet/prodtechnol/winntas/tips/winntmag/inreg.mspx (1997).

[Russinovich 00] Russinovich, Mark. *ListDDLs*. http://www.sysinternals.com/ntw2k/freeware/listdlls.shtml (2000).

[Russinovich 01] Russinovich, Mark. *PsFile*. http://www.sysinternals.com/ntw2k/freeware/psfile.shtml (2001).

[Russinovich 03] Russinovich, Mark. *PsService*. http://www.sysinternals.com/ntw2k/freeware/psservice.shtml (2003).

[Russinovich 04a] Russinovich, Mark. *PsInfo*. http://www.sysinternals.com/ntw2k/freeware/psinfo.shtml (2004).

[Russinovich 04b] Russinovich, Mark & Cogswell, Bryce. *Autoruns*. http://www.sysinternals.com/ntw2k/freeware/autoruns.shtml (2004).

[Russinovich 04c] Russinovich, Mark. *Handle*. http://www.sysinternals.com/ntw2k/freeware/handle.shtml (2004).

[Russinovich 04d] Russinovich, Mark. *Filemon for Windows*. http://www.sysinternals.com/ntw2k/source/filemon.shtml (2004).

[Sorenson 03] Sorenson, Holt. Incident Response Tools For Unix, Part One: System Tools. http://www.securityfocus.com/infocus/1679 (2003).

[Stoffregen 03] Stoffregen, Paul. *Understanding FAT32 Filesystems*. http://www.pjrc.com/tech/8051/ide/fat32.html (2003).

[SystemTools 04] SystemTools.com. *SomarSoft Utilities*. (Click Free Tools.) http://www.somarsoft.com (2004).

[Uniblue 04] Uniblue. *WinTasks Process Library*. http://www.liutilities.com/products/wintaskspro/processlibrary (2004).

[USDOJ 02] United States Department of Justice. *Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations*. http://www.cybercrime.gov/s&smanual2002.htm (2002).

[Vidstrom 04] Vidstrom, Arne. *NETSECURITY.NU*. http://www.ntsecurity.nu/toolbox/macmatch (2004).

[Wheeler 03] Wheeler, David A. *Program Library HOWTO*. http://www.linux.com/howtos/Program-Library-HOWTO/index.shtml (2003).

[Wikipedia 04] Wikipedia’s *Library (computer science)*. http://www.factindex.com/l/li/library_computer_science_html (2004).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE March 2005	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE First Responders Guide to Computer Forensics	5. FUNDING NUMBERS F19628-00-C-0003
6. AUTHOR(S) Richard Nolan, Colin O'Sullivan, Jake Branson, Cal Waits	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI--2005-HB-001
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS	12B DISTRIBUTION CODE 216

13. ABSTRACT (MAXIMUM 200 WORDS)

This handbook is for technical staff members charged with administering and securing information systems and networks. It targets a critical training gap in the fields of information security, computer forensics, and incident response: performing basic forensic data collection. The first module describes cyber laws and their impact on incident response. The second module builds understanding of file systems and outlines a best practice methodology for creating a trusted first responder tool kit for investigating potential incidents. The third module reviews some best practices, techniques, and tools for collecting volatile data from live Windows and Linux systems. It also explains the importance of collecting volatile data before it is lost or changed. The fourth module reviews techniques for capturing persistent data in a forensically sound manner and describes the location of common persistent data types. Each module ends with a summary and a set of review questions to help clarify understanding.

This handbook was developed as part of a larger project. The incorporated slides are from the five day hands-on course Forensics Guide to Incident Response for Technical Staff developed at the SEI. The focus is on providing system and network administrators with methodologies, tools, and procedures for applying fundamental computer forensics when collecting data on both a live and a powered off machine. A live machine is a machine that is currently running and could be connected to the network. The target audience includes system and network administrators, law enforcement, and any information security practitioners who may find themselves in the role of first responder. The handbook should help the target audience to

- understand the essential laws that govern their actions
- understand key data types residing on live machines
- evaluate and create a trusted set of tools for the collection of data
- collect, preserve, and protect data from live and powered off machines
- learn methodologies for collecting information that are forensically sound (i.e., able to withstand the scrutiny of the courts)

14. SUBJECT TERMS computer forensics, information security, data recovery, computer security incident, cyber law	15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL